(1)

CRL COMPUTING RESEARCH LABORATORY

Memoranda in Computer and Cognitive Science

New Mexico State University

Proceedings of the First Workshop on Proximity Graphs

D. W. Dearholt and F. Harary

MCCS-91-224

Computing Research Laboratory
Box 30001
New Mexico State University
Las Cruces, New Mexico 88003

ABSTRACT

The motivations for holding this workshop came from the recently discovered associations between proximity graphs and Pathfinder networks. The elegant theoretical domain and the breadth of applications make this a very rich area indeed. The workshop was attended by several of the leading researchers in proximity graphs, and was organized so that there would be adequate opportunity for discussion of common interests. The presentations were organized into four sections: theoretical foundations, algorithms and computational aspects, applications, and graphics and unsolved problems. There were also demonstrations of three systems based on proximity graphs: information retrieval using Pathfinder networks, a robotic vision database system organized as a monotonic search network, and a UNIX help system on a Hypertext Browser organized as a Pathfinder network. A tool to display and manipulate large graphs was also demonstrated. The workshop brought together some mainstream graph theorists and the researchers who had been working on proximity graphs as a special case of graph theory, and the interchange was profitable for all.

The Computing Research Laboratory was established by the New Mexico State Legislature, under the Science and Technology Commercialization Commission as part of the Rio Grande Research Corridor.

This material is based upon work supported by the National Science Foundation under Grant No. CCR-8918729.

This work relates to Department of Navy Grant N00014-90-J-1323 issued by the Office of Naval Research. The United States Government has a royalty-free license throughout the world in all copyrightable material contained herein.

Proceedings of The First Workshop on Proximity Graphs

November 30 through December 2, 1989

Las Cruces, New Mexico

D. W. Dearholt and F. Harary, Organizers

Sponsored by:

The National Science Foundation

The Office of Naval Research

and

New Mexico State University:

The Department of Computer Science

and

The Computing Research Laboratory

PREFACE

The motivations for holding this workshop came from my recent discovery of the research on proximity graphs, by means of my own studies in the development of Pathfinder networks. The fascinating theoretical associations which have come from the work on proximity graphs, and the breadth of applications of them, make this a very rich area indeed. The connections with computational geometry and complexity theory, the modeling of aspects of vision and perception, and of human associative memory, all contribute to the promise of the perspective afforded by proximity graphs. In my early work (as evidenced by the transparencies herein), I had named these graphs *empty-neighborhood graphs*, with the intent of generalizing the concept beyond the well-known cases which had been studied, before realizing that others had called them *proximity graphs*. But the name itself is of relatively little importance; the domain is the thing, and it is a veritable banquet of interesting problems and applications.

The workshop was designed to be as informal as possible, and the intention was to have the contributors present their most recent work; thus no formal papers were expected, and the proceedings consist of abstracts and transparencies. Unfortunately, the editor has moved in the intervening time, and this has caused the delay in the final organization and distribution of these proceedings; I offer my apologies to all.

I would like to express my grateful appreciation to those who helped make this workshop possible: those who came from many places and presented their work, and participated in the many discussions; to Kamal Abdali of NSF, and Marc Lipman of ONR, who encouraged the workshop with their moral support, and who also aided in obtaining financial support; to Marc Lipman for attending and moderating the panel discussion; to the graduate students at NMSU, who helped in many ways in preparing for the workshop, in giving demonstrations and in acting as hosts for the visitors; to my research colleagues at NMSU--particularly Ken Paap, Roger Schvaneveldt, Jim McDonald, Art Knoebel, and Keith Phillips--with whom many profitable discussions were held; to the Department of Computer Science and the College of Arts and Sciences at NMSU, for their support and encouragement; and to Frank Harary, whose insights in both graph theory and workshops helped in numerous ways.

Don Dearholt
Mississippi State University
May 22, 1991

TABLE OF CONTENTS

The First Workshop on Proximity Graphs

ABSTRACT

Introduction to Pathfinder Networks (PFNETs) and Relationships Between Them and Proximity Graphs

Donald W. Dearholt
Department of Computer Science
Mississippi State University
MS 39762

A graph model for human semantic memory (named "Pathfinder") has proven to be a rich source of associations with both graph theory and computational geometry, including *proximity graphs*. New theorems and heuristics have been devised based upon the properties of Pathfinder networks, and a new class of proximity graphs has been defined. Relationships with previously studied proximity graphs have also been established. Applications resulting from the research in Pathfinder networks include online help systems, a hypertext browser, and a vision database intended for robotics applications. These applications feature the organization of information according to the principles of organization of Pathfinder networks. As such, these applications support various levels of abstraction and clustering, and principled associations between clusters. The advantages of using Pathfinder-based networks in the human-computer interface or a specialized database include (1) the provision for a good "cognitive match" with users, (2) higher levels of abstraction and clustering are supported, (3) the organization is typically nonhierarchical, allowing multiple paths of access to needed information, and (4) the most salient relationships (often the most frequently used paths) are represented explicitly as edges in the network.

OUTLINE

I. MOTIVATION, PERSPECTIVE, AND OBJECTIVES

II. PATHFINDER NETWORKS

A. DEFINITIONS AND PROPERTIES

B. APPLICATIONS

III. EMPTY-NEIGHBORHOOD (PROXIMITY) GRAPHS

A. DEFINITIONS AND PROPERTIES

B. APPLICATIONS

PHILOSOPHICAL STANCE: BETTER MODELING OF HUMAN INTELLIGENCE WILL LEAD TO BETTER AI

THE NETWORKS WE ARE STUDYING:

DESCRIBE, SUMMARIZE, AND DISPLAY DATA

SUGGEST A PSYCHOLOGICAL MODEL ABOUT MENTAL REPRESENTATIONS

COMPLEMENT MDS AND CLUSTER ANALYSIS

PROVIDE A PARADIGM FOR:

- KNOWLEDGE REPRESENTATION
- MODELS OF CLASSIFICATION
- ORGANIZATION OF DATABASE SYSTEMS
- SPREADING ACTIVATION (SEARCH)

THE BIGGEST CHALLENGE

FOR AI AND COGNITIVE MODELING:

TO DESIGN A SYSTEM WHICH DOES MANY THINGS WELL, ALTHOUGH EACH ALGORITHM MIGHT NOT BE OPTIMAL

- ASSOCIATIONAL ORGANIZATION
- CLUSTERING
- SEVERAL LEVELS OF ABSTRACTION
- CLASSIFICATION
- SEARCH
- DESCRIPTION OF DECISIONS

RESEARCH OBJECTIVES

I. THEORETICAL

DEVELOP AND TEST METRICS

RELATIONSHIPS:
- GRAPH THEORY
- PATH ALGEBRAS
- PROXIMITY GRAPHS (RNG, GG, DTG)

LEVELS OF ABSTRACTION

II. EMPIRICAL

SEMANTIC MEMORY

CLASSIFICATION MODELS

PROPOSITIONAL ANALYSIS

KNOWLEDGE EXTRACTION FROM EXPERTS

III. APPLICATION DOMAINS

ORGANIZATION OF CONCEPTS
- INTERFACES--INFORMATION RETRIEVAL, HELP SYSTEMS
- DATABASE ORGANIZATION

PERCEPTION--OUTLINES OF OBJECTS

WHY USE GRAPHS?

ALTERNATIVES:

MULTIDIMENSIONAL SCALING

CLUSTERING

CORRELATION MATRIX

GRAPHS HAVE BEEN USED FOR:

KNOWLEDGE REPRESENTATION

CONCEPT LEARNING

MODELS OF SEMANTIC MEMORY

ORGANIZATION OF A DATABASE SYSTEM

ASSOCIATIVE SEARCH

CLASSIFICATION

DESCRIPTION AND DECISIONS

WHAT IS SIMILARITY?

I. SUBJECTIVE

$$A \left\{ \begin{array}{l} \text{IS LIKE} \\ \text{IS SIMILAR TO} \\ \text{RESEMBLES} \\ \text{IS A KIND OF} \\ \text{PARALLELS} \end{array} \right\} B$$

PURPOSE: CONFERS PROPERTIES OF B UPON A

ONLY ESTIMATES ARE AVAILABLE

CAN BE ASYMMETRIC

RESULT OF DIFFERENT ASSOCIATIONS/FEATURES

II. OBJECTIVE

L1 OR L2 NORM (DISTANCE)

$$\frac{\text{SET INTERSECTION}}{\text{SET UNION}} \quad \text{(CO-OCCURRENCE)}$$

EXAMPLES OF ASYMMETRIC SIMILARITY

NORTH KOREA IS LIKE CHINA

THE PORTRAIT RESEMBLES YOU

TRUE LOVE IS AS DEEP AS THE OCEAN

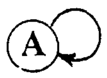
LIFE IS LIKE A PLAY

A PLAY IS LIKE LIFE

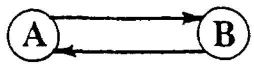
THE METRIC AXIOMS

GIVEN ENTITIES A, B, AND C:

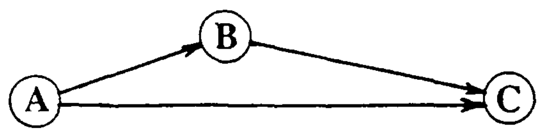
1. THE DISTANCE BETWEEN AN ENTITY AND ITSELF IS ZERO



2. THE DISTANCE FROM A TO B IS THE SAME AS THE DISTANCE FROM B TO A



3. THE DISTANCE FROM A TO C IS LESS THAN OR EQUAL TO THE DISTANCE FROM A TO B AND THEN TO C



DEFINITION

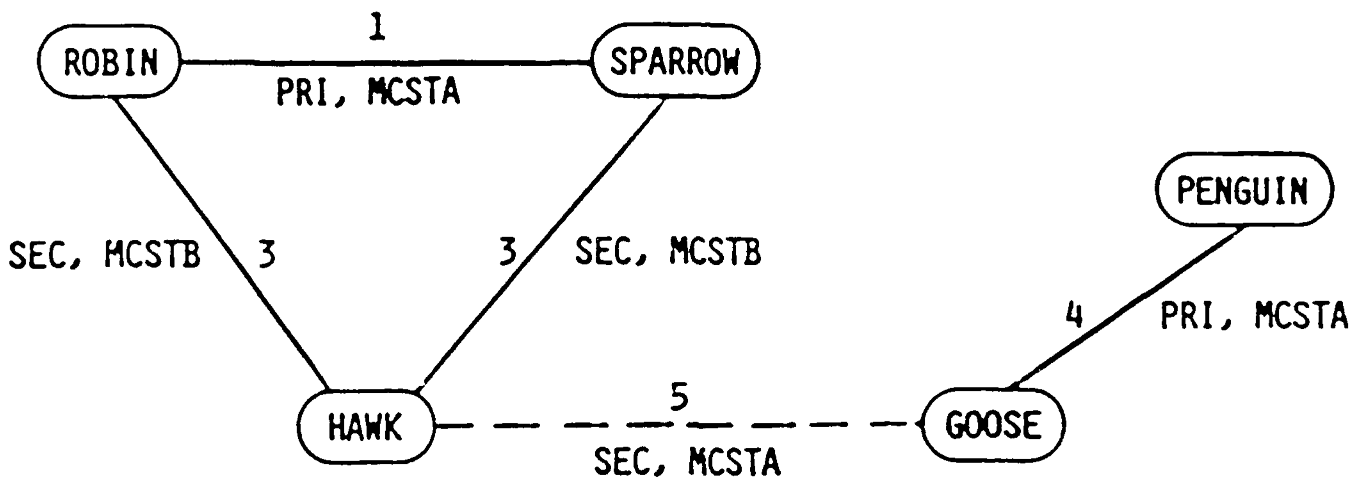
A PATHFINDER NETWORK (PFNET) IS A GRAPH BASED ON PAIRWISE ESTIMATES OR MEASURES OF DISTANCES BETWEEN ENTITIES.

EACH ENTITY CORRESPONDS TO A NODE.

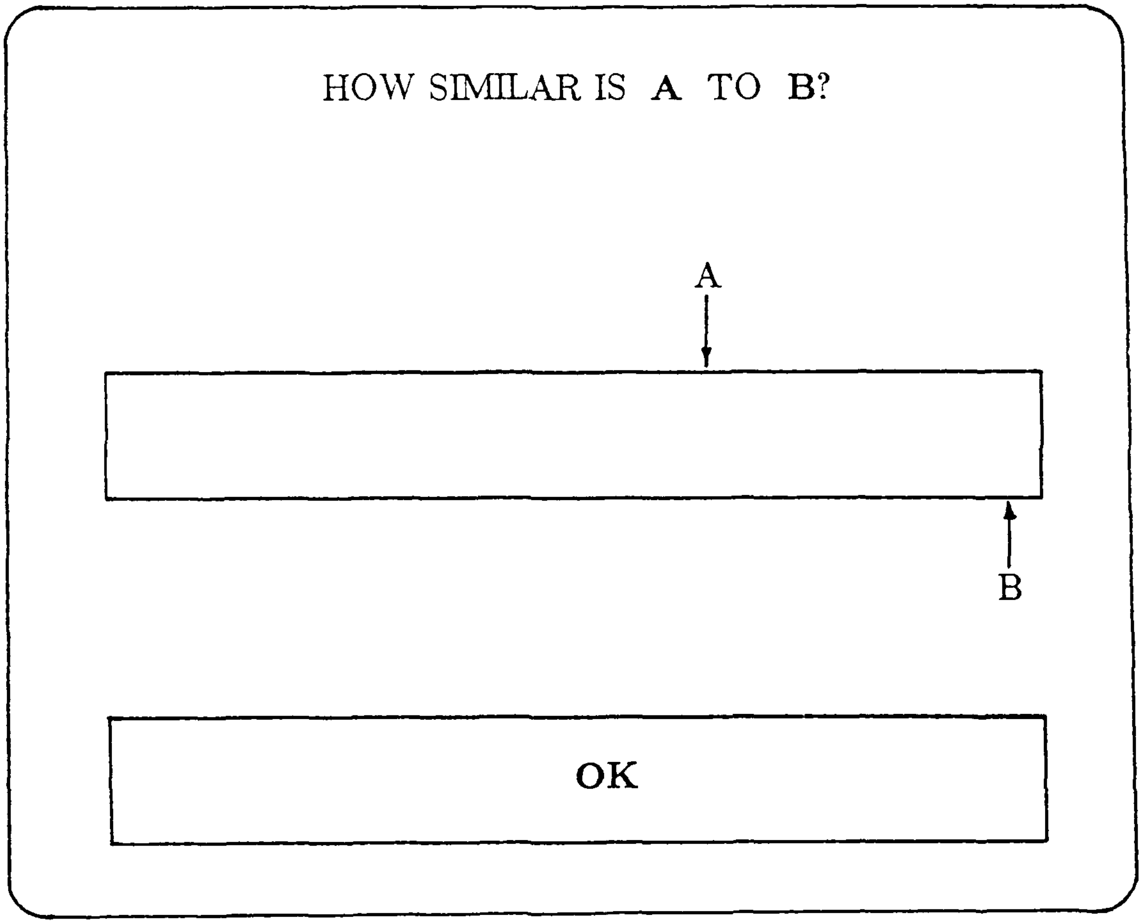
EACH PAIR OF NODES IN A PFNET IS CONNECTED DIRECTLY BY AN EDGE WHOSE WEIGHT IS THE DISTANCE BETWEEN THE TWO ENTITIES,

UNLESS THERE IS A SHORTER ALTERNATIVE PATH.

EXAMPLE OF A LABELED PFNET



TOUCHSCREEN DISPLAY FOR EMPIRICAL DATA



$$DISTANCE + SIMILARITY = K$$

THE PARAMETERS OF A PFNET

R-METRIC:

RULE FOR FINDING THE LENGTH OF A PATH WITH K EDGES

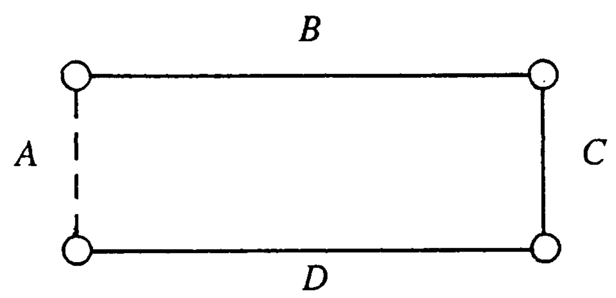
$$L(P) = \left[\sum_{I=1}^{K} W_I^{R} \right]^{1/R}$$

R	PATH LENGTH	DATA SCALE
1	SUM OF WEIGHTS	RATIO
2	EUCLIDEAN	RATIO
•		
•		
∞	MAXIMUM WEIGHT	RATIO, ORDINAL

THE PARAMETERS OF A PFNET

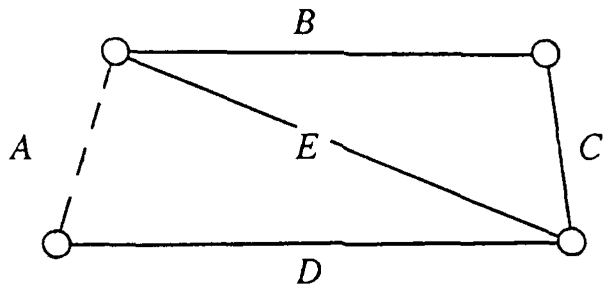
Q-PARAMETER:

"DIMENSION" OF GENERALIZED TRIANGLE INEQUALITIES SATISFIED



$$A \le \left[B^R + C^R + D^R\right]^{1/R}$$

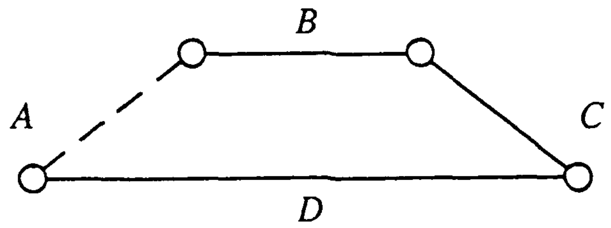
THE TRIANGLE INEQUALITY



$E \leq B + C$

$A \leq E + D \leq B + C + D$

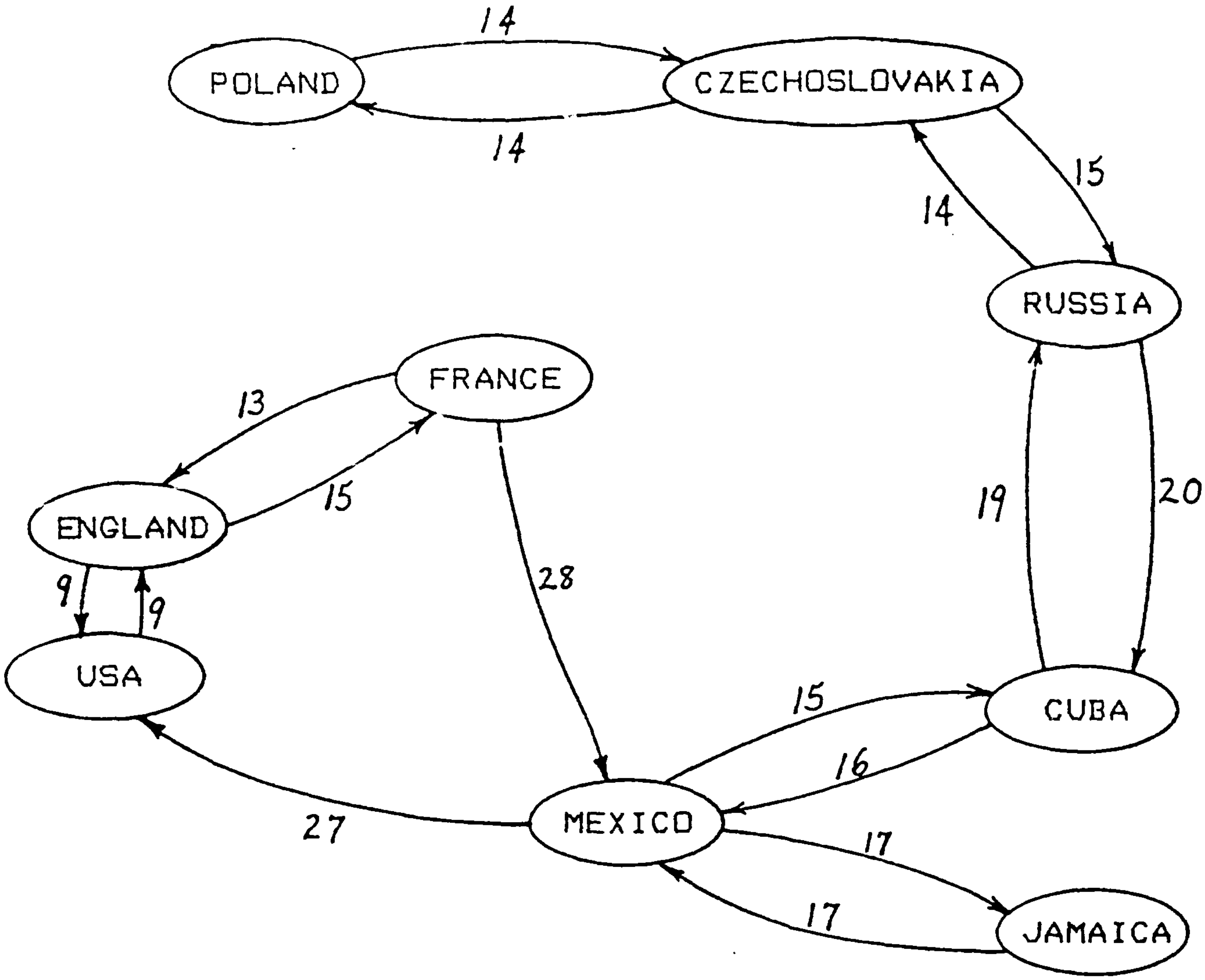
THE GENERALIZED TRIANGLE INEQUALITY



$A \leq \left[B^R + C^R + D^R\right]^{1/R}$

PURPOSE: TO PRESERVE MINIMAL-DISTANCE PATHS

DIRECTED PFNET FOR NINE COUNTRIES



R-METRIC IS INFINITY

Q-PARAMETER IS EIGHT

THEORETICAL RESULTS

FOR A GIVEN DISTANCE MATRIX,

PFNET(R,Q):

IS UNIQUE,

PRESERVES GEODETIC DISTANCES,

LINKS NEAREST NEIGHBORS, AND

CONTAINS THE SAME INFORMATION AS THE MINIMUM METHOD OF HIERARCHICAL CLUSTERING

PFNET($R=\infty$,$Q=N-1$) IS THE UNION OF ALL MINTREES

PFNET(R_2,Q) IS A SPANNING SUBGRAPH OF PFNET(R_1,Q) IFF $R_1 \leq R_2$

PFNET(R,Q_2) IS A SPANNING SUBGRAPH OF PFNET(R,Q_1) IFF $Q_1 \leq Q_2$

MONOTONIC TRANSFORMATIONS PRESERVE STRUCTURE FOR ALL PFNET($R=\infty$,Q)

MULTIPLICATIVE TRANSFORMATIONS PRESERVE STRUCTURE FOR ALL PFNET(R, Q)

OPEN PROBLEMS

CLASSIFICATION

- METRICS
- STRUCTURE
  - EDGE LABELS
  - STABILITY
  - LEVELS OF ABSTRACTION

GRAPHICAL REPRESENTATIONS

SEARCH

- SPECIALIZED DATABASES
- SPREADING ACTIVATION
- MATCH CRITERION

EXPLOITING PARALLELISM

- SEARCH
- CLASSIFICATION

APPLICATIONS FOR PATHFINDER-BASED ASSOCIATIVE NETWORKS

I. INTERFACE DESIGN

HYPERTEXT BROWSER (HYBROW)

DOMAINS: UNIX CONSULTANT, INCIDENT DATABASE

INFORMATION RETRIEVAL (PATHTRIEVE)

DOMAIN: ABSTRACTS OF DOCUMENTS

MEASURE OF PROXIMITY: CO-OCCURRENCE OF CONCEPTS

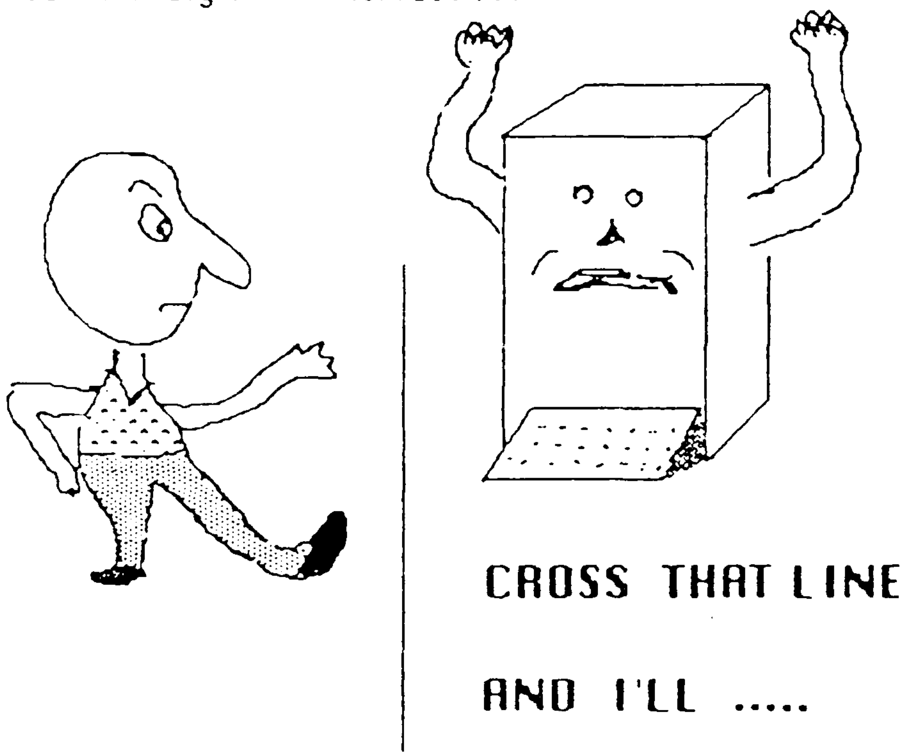
II. DATABASE ORGANIZATION

ROBOTIC VISION SYSTEM

DOMAIN: FOURIER VECTORS OF OUTLINES OF OBJECTS

MEASURE OF PROXIMITY: L2 NORM DISTANCE

The Typical Friendly User Interface ....



T H E I

U N

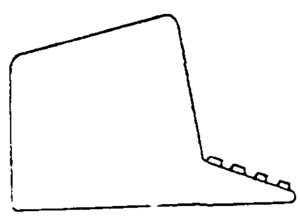
C O M P U T E R

A E

N R

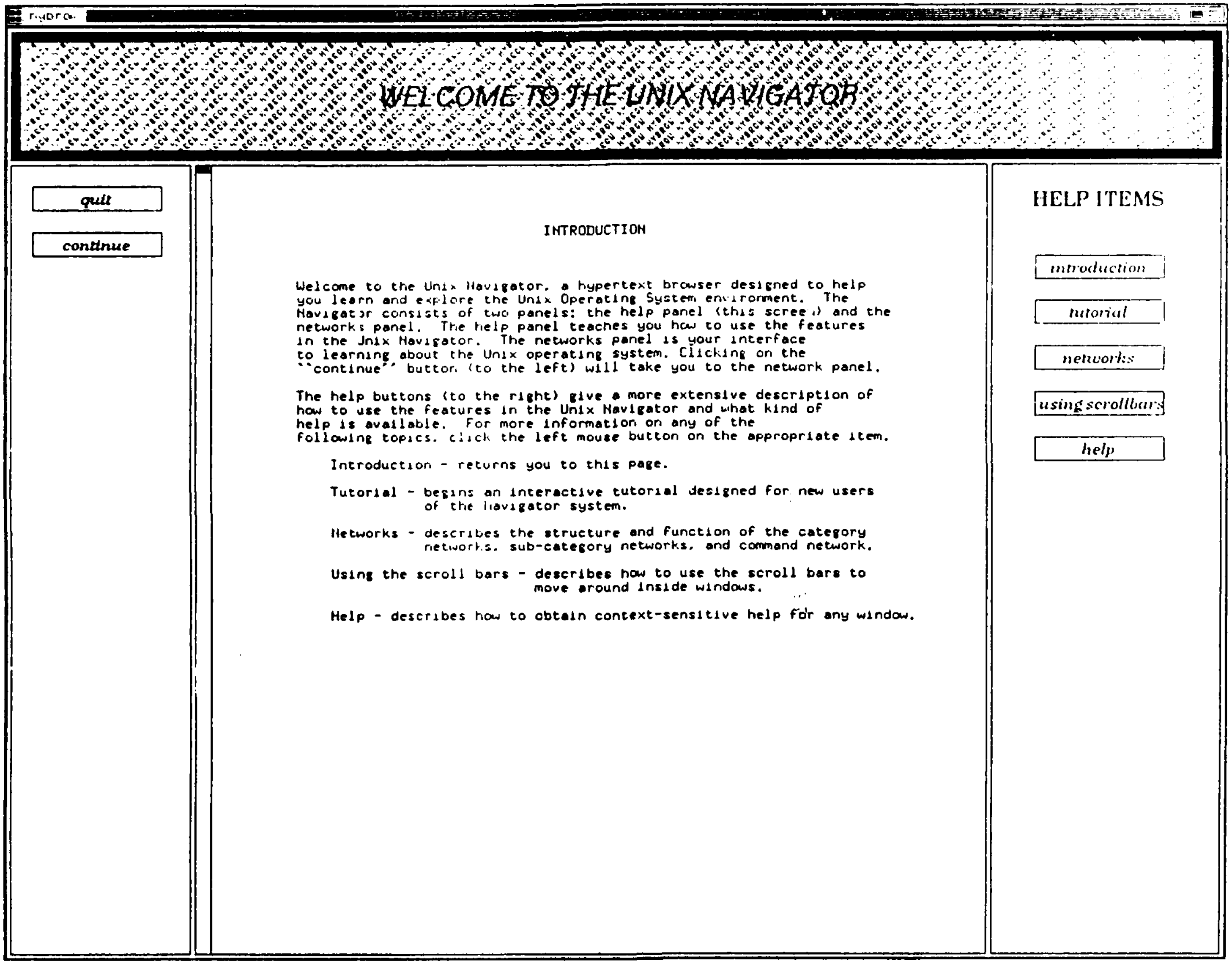
F

A



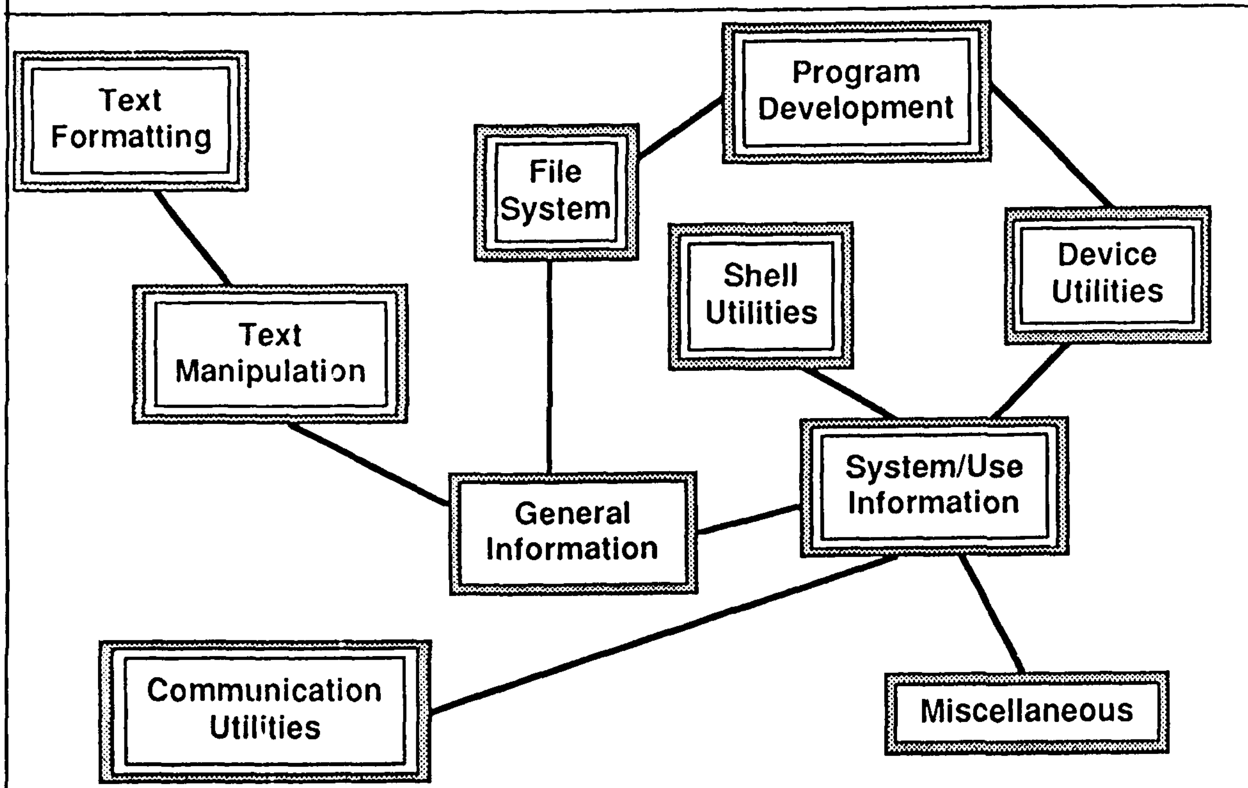
C

E


WELCOME TO THE UNIX NAVIGATOR
quit
continue
INTRODUCTION
Welcome to the Unix Navigator, a hypertext browser designed to help you learn and explore the Unix Operating System environment. The Navigator consists of two panels: the help panel (this screen) and the networks panel. The help panel teaches you how to use the features in the Unix Navigator. The networks panel is your interface to learning about the Unix operating system. Clicking on the ``continue'' button (to the left) will take you to the network panel.
The help buttons (to the right) give a more extensive description of how to use the features in the Unix Navigator and what kind of help is available. For more information on any of the following topics, click the left mouse button on the appropriate item.
Introduction - returns you to this page.
Tutorial - begins an interactive tutorial designed for new users of the Navigator system.
Networks - describes the structure and function of the category networks, sub-category networks, and command network.
Using the scroll bars - describes how to use the scroll bars to move around inside windows.
Help - describes how to obtain context-sensitive help for any window.
HELP ITEMS
introduction
tutorial
networks
using scrollbars
help

TOP CATEGORY NETWORK

Synopsis: This is a network of the top-level categories. Click on any box to see the subcategories nested within any category of interest.

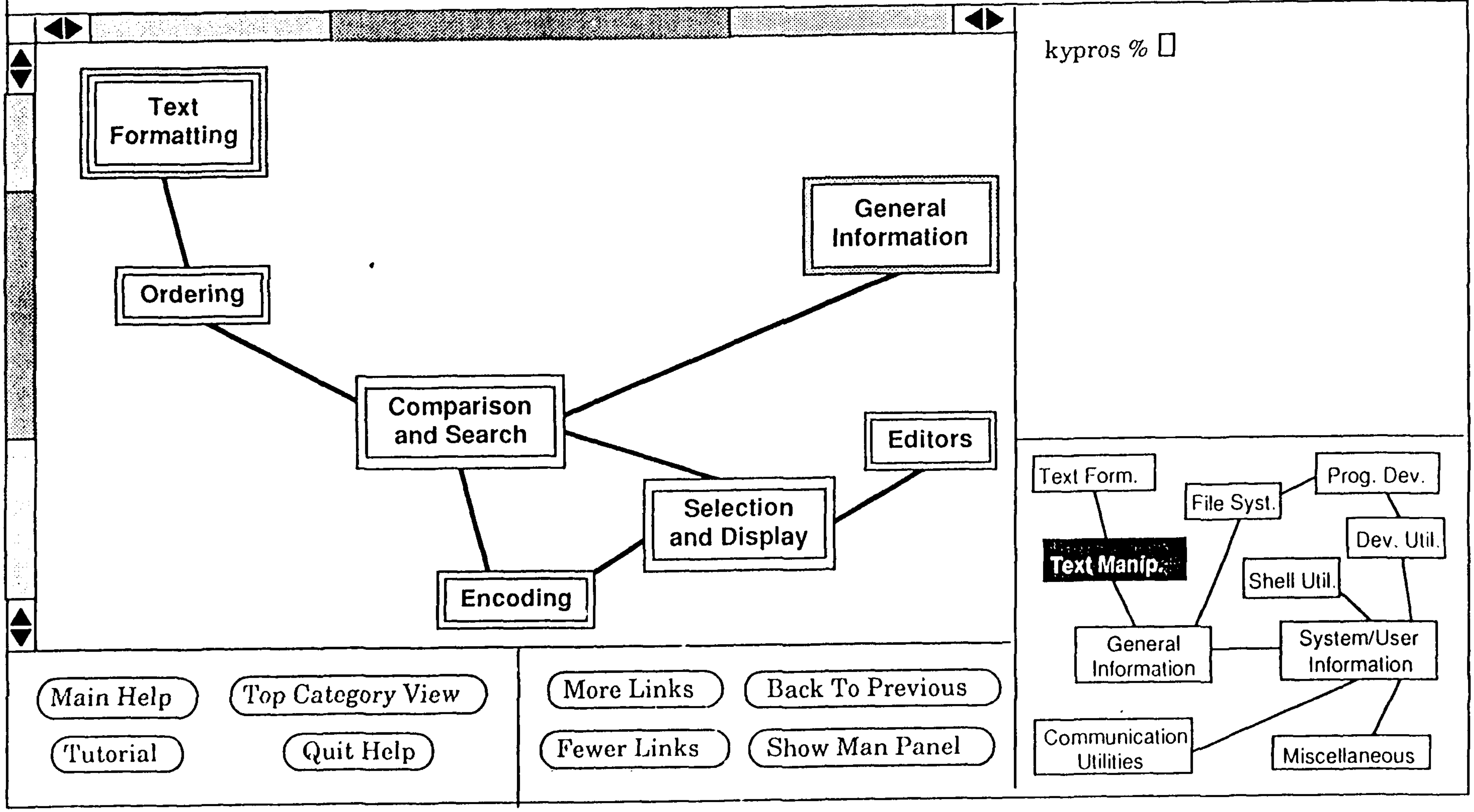


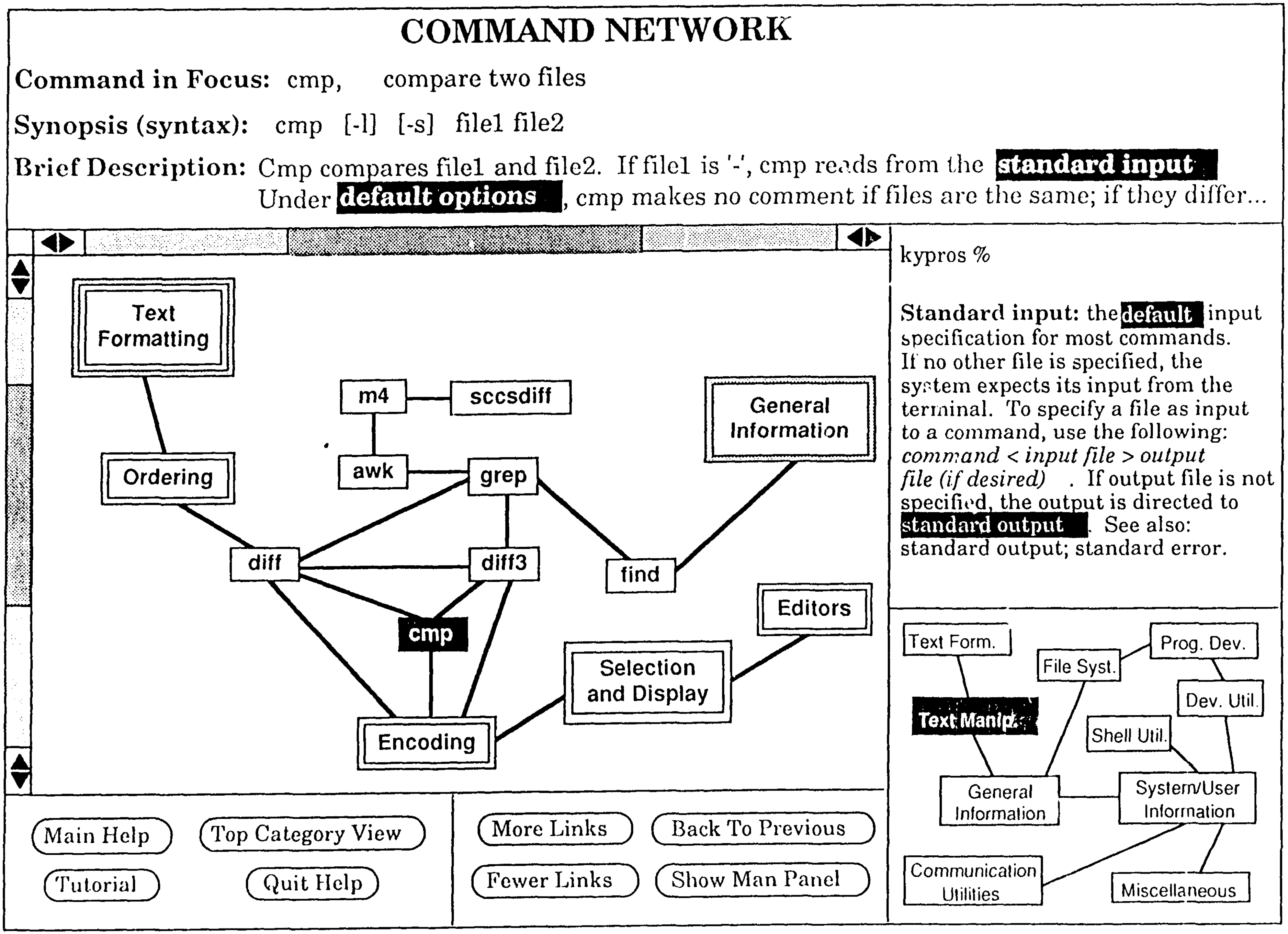
kypros %

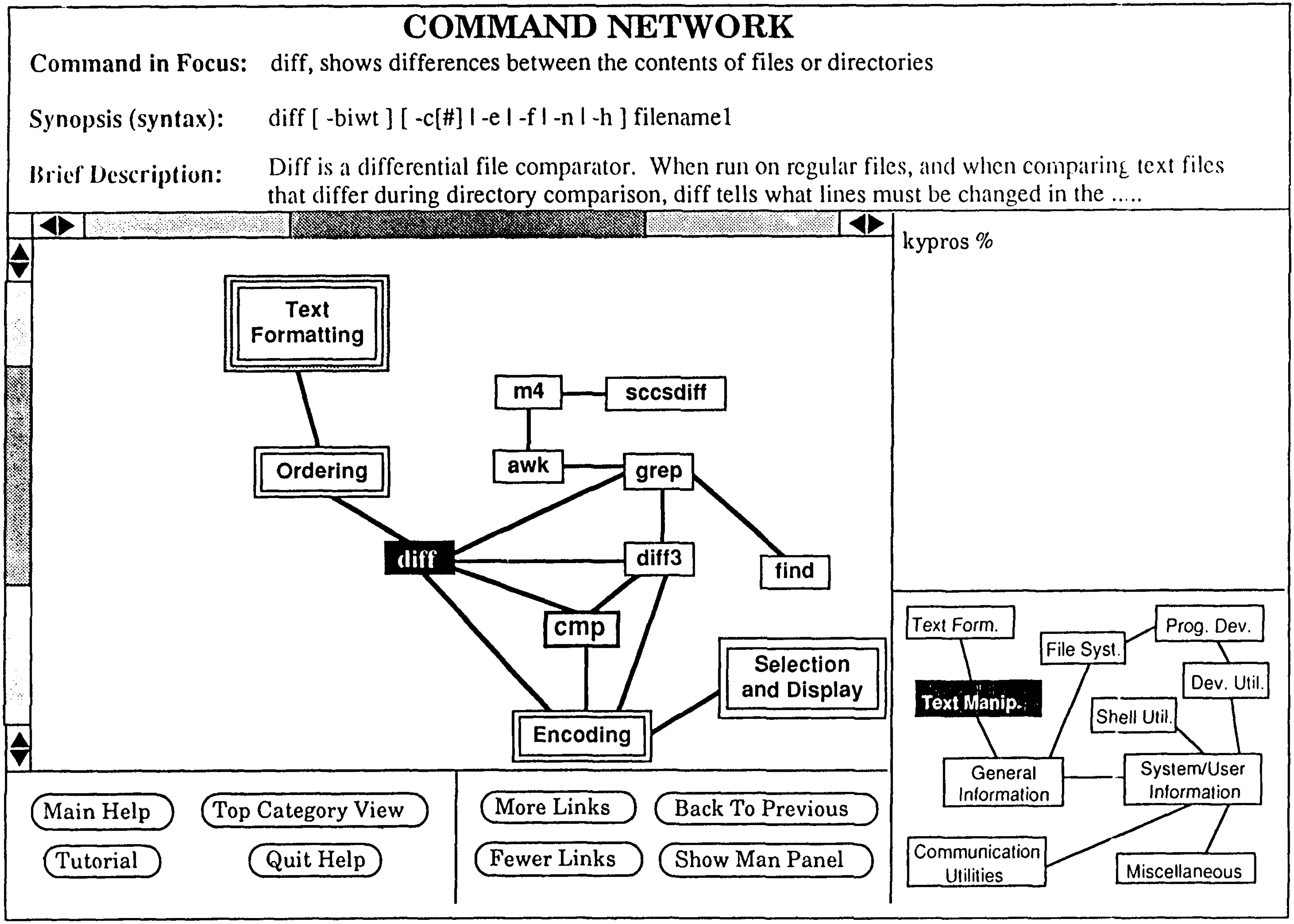
Main Help
Top Category View
Tutorial
Quit Help
More Links
Back To Previous
Fewer Links
Show Man Panel

COMMAND NETWORK

Synopsis: *The text manipulation category has been opened.* Click on any of the five subcategories to see the commands nested within that subcategory.




COMMAND NETWORK
Command in Focus: cmp, compare two files
Synopsis (syntax): cmp [-l] [-s] file1 file2
Brief Description: Cmp compares file1 and file2. If file1 is '-', cmp reads from the standard input
Under default options, cmp makes no comment if files are the same; if they differ...
Text Formatting
Ordering
m4
sccsdiff
awk
grep
diff
diff3
cmp
find
General Information
Editors
Selection and Display
Encoding
Main Help
Top Category View
Tutorial
Quit Help
More Links
Back To Previous
Fewer Links
Show Man Panel
kypros %
Standard input: the default input specification for most commands. If no other file is specified, the system expects its input from the terminal. To specify a file as input to a command, use the following: command < input file > output file (if desired) . If output file is not specified, the output is directed to standard output. See also: standard output; standard error.
Text Form.
File Syst.
Prog. Dev.
Dev. Util.
Text Manip.
Shell Util.
General Information
System/User Information
Communication Utilities
Miscellaneous


COMMAND NETWORK
Command in Focus: diff, shows differences between the contents of files or directories
Synopsis (syntax): diff [-biwt] [-c[#] | -e | -f | -n | -h] filename1
Brief Description: Diff is a differential file comparator. When run on regular files, and when comparing text files that differ during directory comparison, diff tells what lines must be changed in the .....
kypros %
Text Formatting
Ordering
diff
m4
sccsdiff
awk
grep
diff3
find
cmp
Encoding
Selection and Display
Text Form.
File Syst.
Prog. Dev.
Dev. Util.
Text Manip.
Shell Util.
General Information
System/User Information
Communication Utilities
Miscellaneous
Main Help
Top Category View
Tutorial
Quit Help
More Links
Back To Previous
Fewer Links
Show Man Panel

COMPUTER VISION

GOAL: SCAN THE ENVIRONMENT AND MAKE DECISIONS WITHOUT HUMAN INTERACTION

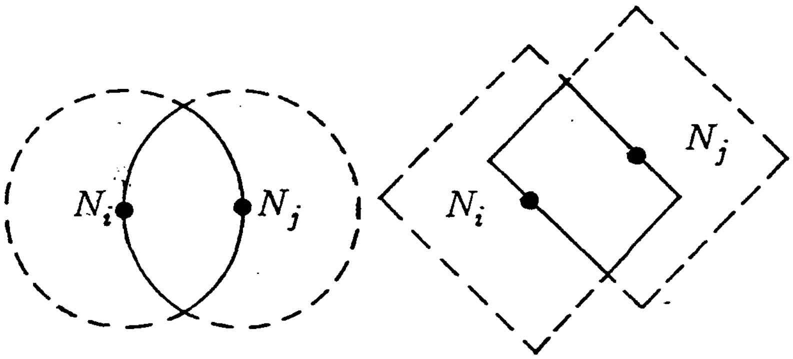
REQUIRES: KNOWLEDGE REPRESENTATION
CLASSIFICATION
ABILITY TO *DESCRIBE* SCENE
RECONSTRUCT SCENE
ENHANCE SCENE
MODIFY SCENE

DEFINITION OF AN EMPTY-NEIGHBORHOOD GRAPH

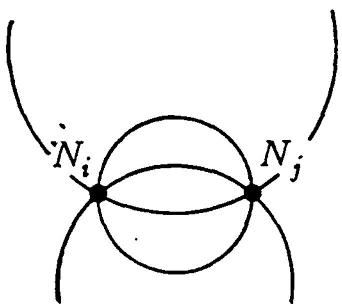
An ENG is a graph in which a link l_{ij} is in the ENG if and only if an open neighborhood associated with N_i and N_j is empty of all other nodes (points).

If each pair of nodes determines a unique neighborhood, then the graph is referred to as a **single-neighborhood** ENG.

If each pair of nodes determines a set of possible neighborhoods, then the graph is referred to as a **family-neighborhood** ENG.

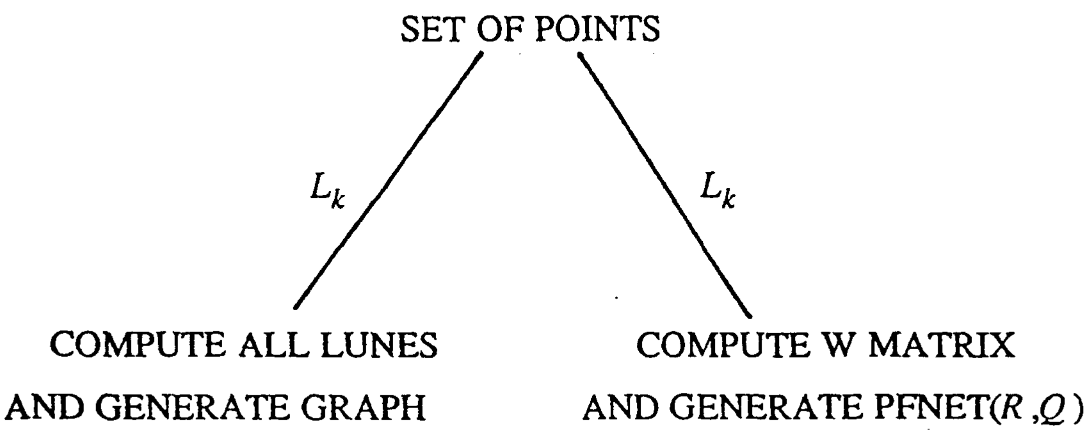


SINGLE NEIGHBORHOODS



A FAMILY NEIGHBORHOOD

GENERATION OF A PROXIMITY GRAPH



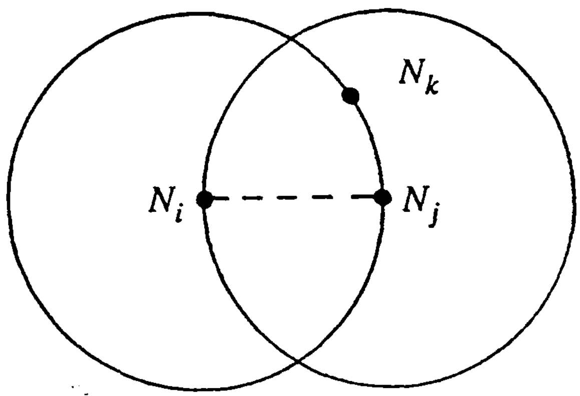
RELATIVE NEIGHBORHOOD GRAPHS

DEFINITION: N_i AND N_j ARE LINKED IN $RNG(L_2)$ IFF

$$d(N_i, N_j) \leq d(N_i, N_k)$$

OR

$$d(N_i, N_j) \leq d(N_j, N_k) \quad \text{FOR ALL } N_k$$



PATHFINDER NETWORKS

DEFINITION: N_i AND N_j ARE LINKED IN $PFNET(L_2, \infty, 2)$ IFF

$$d(N_i, N_j) \leq \text{MIN}[\ \text{MAX}[\ d(N_i, N_k), d(N_j, N_k)]]$$

FOR ALL N_k

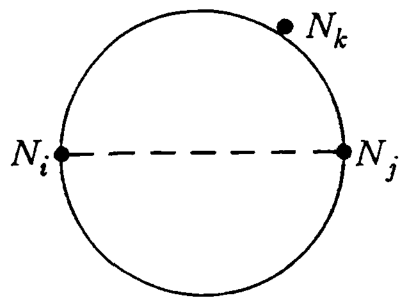
THEREFORE $RNG(L_2) = PFNET(L_2, \infty, 2)$

GABRIEL GRAPHS

DEFINITION: N_i AND N_j ARE LINKED IN $GG\,(L_2)$ IFF

$$d_{ij} < [d_{ik}{}^2 + d_{kj}{}^2]^{1/2}$$

FOR ALL N_k



PATHFINDER NETWORKS

DEFINITION: N_i AND N_j ARE LINKED IN $PFNET\,(L_2, 2, 2)$ IFF

$$d_{ij} \leq \mathrm{MIN}[\ \mathrm{MAX}[d_{ik}{}^2 + d_{kj}{}^2]^{1/2}]$$

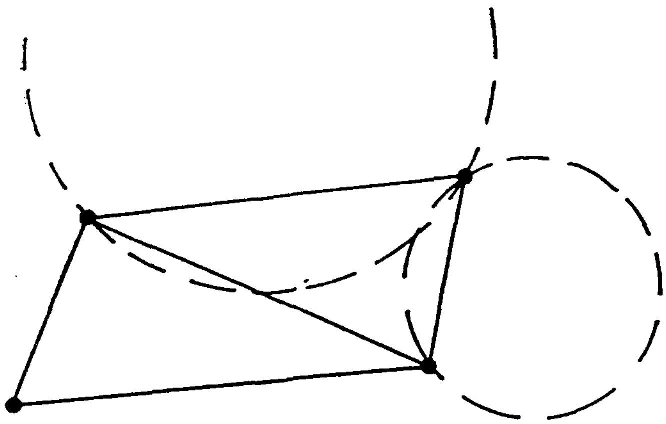
FOR ALL N_k

THEREFORE $MGG\,(L_2) = PFNET\,(L_2,2,2)$

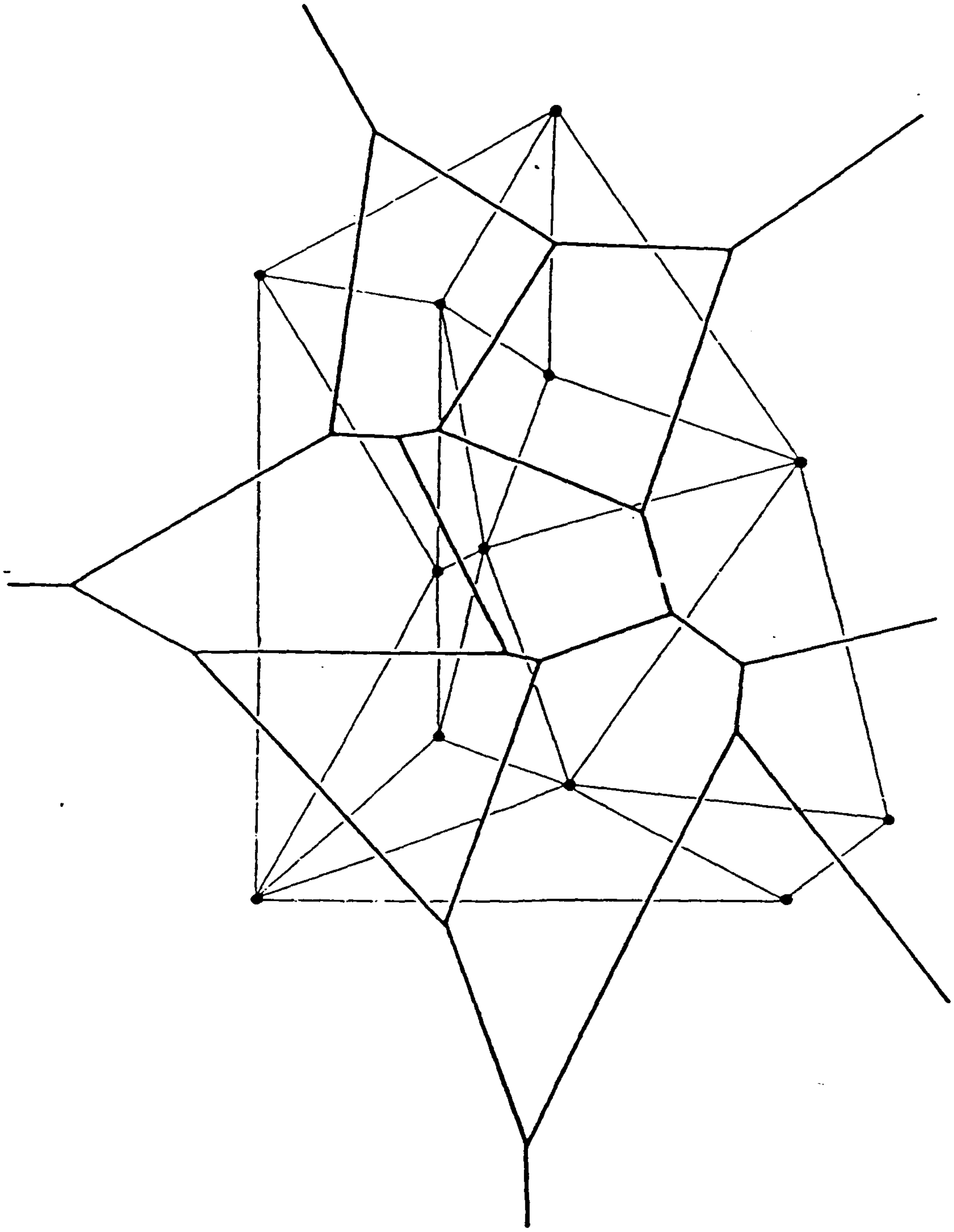
DELAUNAY TRIANGULATION GRAPHS

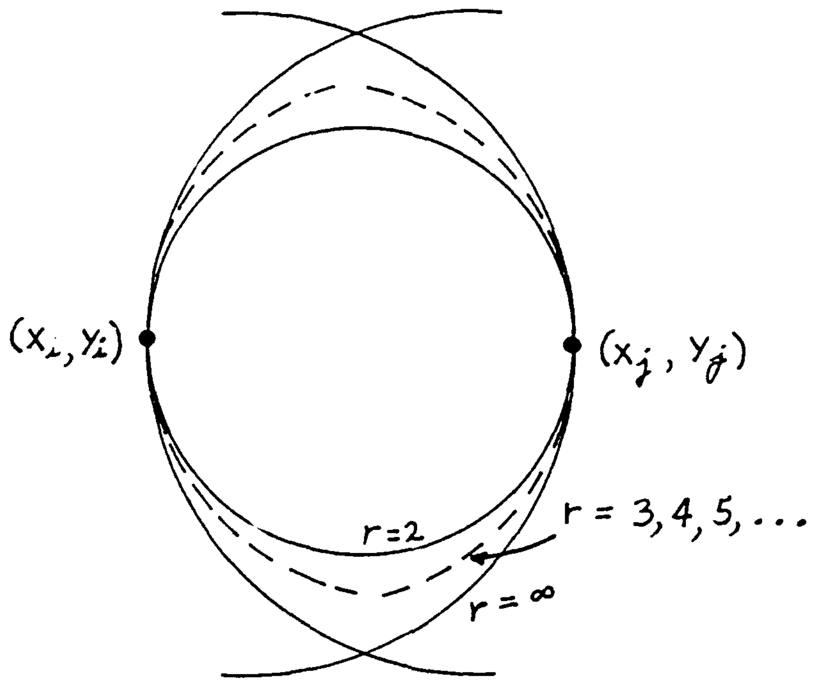
DEFINITION (O'ROURKE): N_i AND N_j ARE LINKED IN $DTG\,(L_k)$ IFF THERE EXISTS AN OPEN BALL B WITH BOUNDARY S SUCH THAT:

1. S PASSES THROUGH N_i and N_j, AND
2. B IS EMPTY



DTG(L_2)



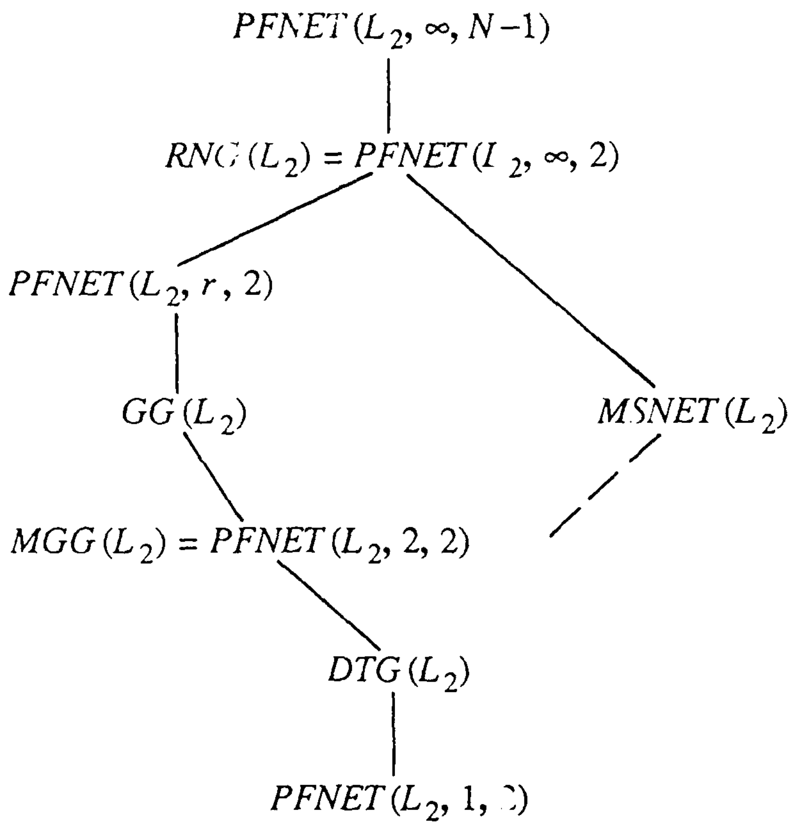


$$d^r = \left[(x_i - x)^2 + (y_i - y)^2\right]^{r/2} + \left[(x_j - x)^2 + (y_j - y)^2\right]^{r/2}$$

THE LUNES OF

PFNET$(L_2, r, 2)$

A HIERARCHY OF EMPTY NEIGHBORHOOD GRAPHS



EACH GRAPH IS A SPANNING SUBGRAPH OF THE GRAPH BELOW IT

Centrality in Proximity Graphs

Fred Buckley
Mathematics Department
Baruch College (CUNY)
New York, NY 10010

Abstract

Over the years many different centrality concepts have been developed. Their most important use in applications has been in facility location problems. In these problems, one typically wants to determine a "good" location for a proposed facility such as a police station, hospital, power station, telecommunications switching center, or a collection of railway depots. Which centrality concept to use depends on the application. We discuss centrality concepts and indicate approaches for their determination in proximity graphs. Recent results for two different types of centers of polygons are also described.

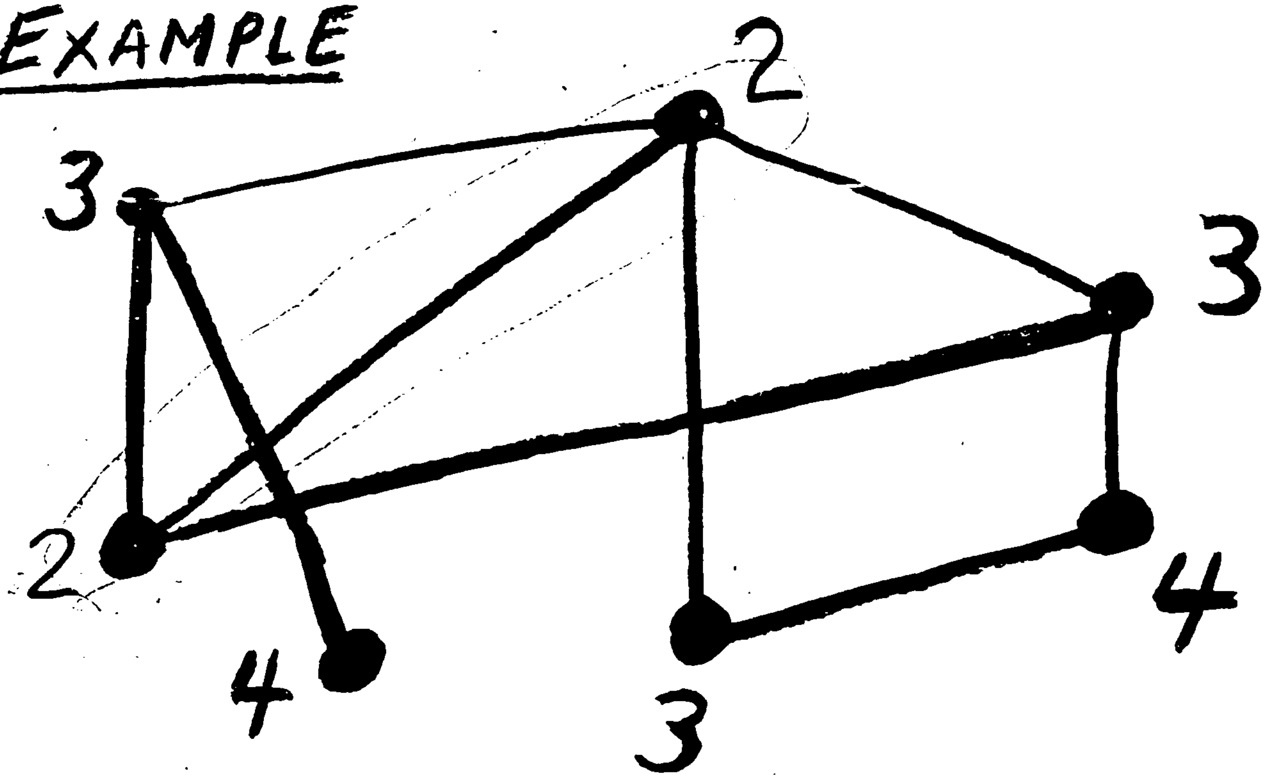
The <u>eccentricity</u> of a vertex v is the distance to a vertex farthest from v.

The <u>center</u> of a graph G is the set of vertices that have minimum eccentricity.

<u>Notation</u> $e(v)$

$C(G)$

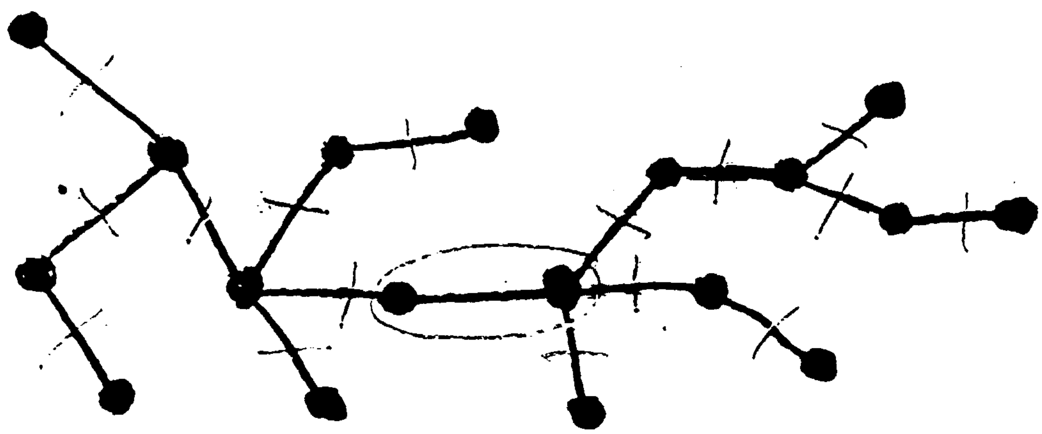
EXAMPLE



We call $\langle C(G) \rangle$ the <u>central</u> <u>subgraph</u> of G.

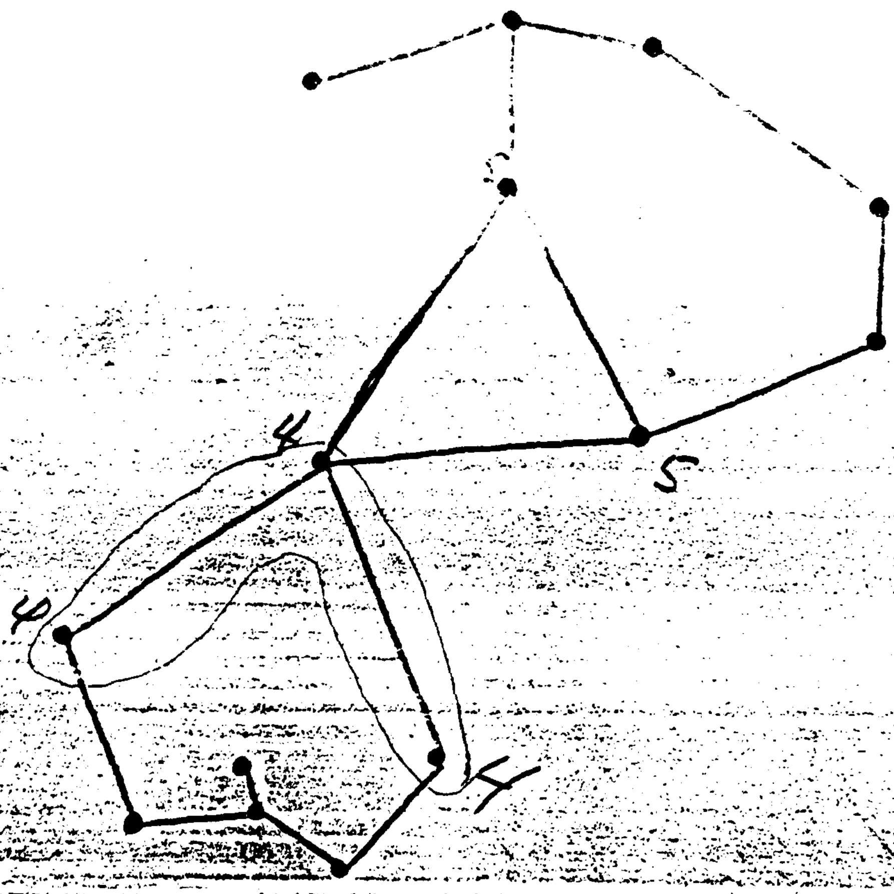
<u>Theorem</u> (Jordan, 1869) If T is a tree, then $\langle C(T) \rangle \cong K_1$ or K_2.

Finding the center of a tree



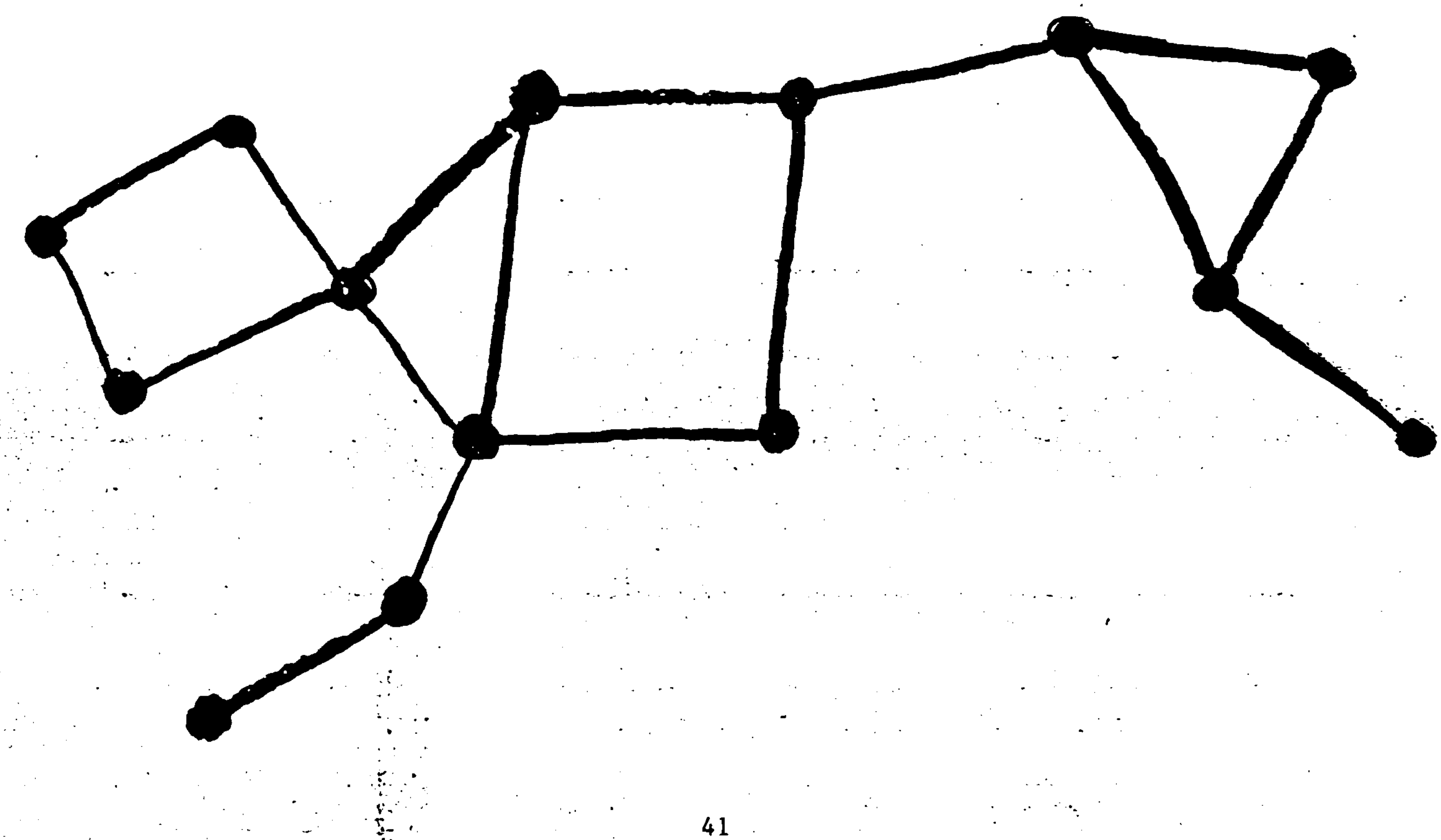
$O(n)$ algorithms for $C(T)$
$O(n^3)$ algorithm for $C(G)$

Given points $p_1, p_2, \ldots, p_n$, a pair p_i, p_j are **relative neighbors** if $d(p_i, p_j) \le \max\{d(p_i, p_k), d(p_j, p_k)\}$



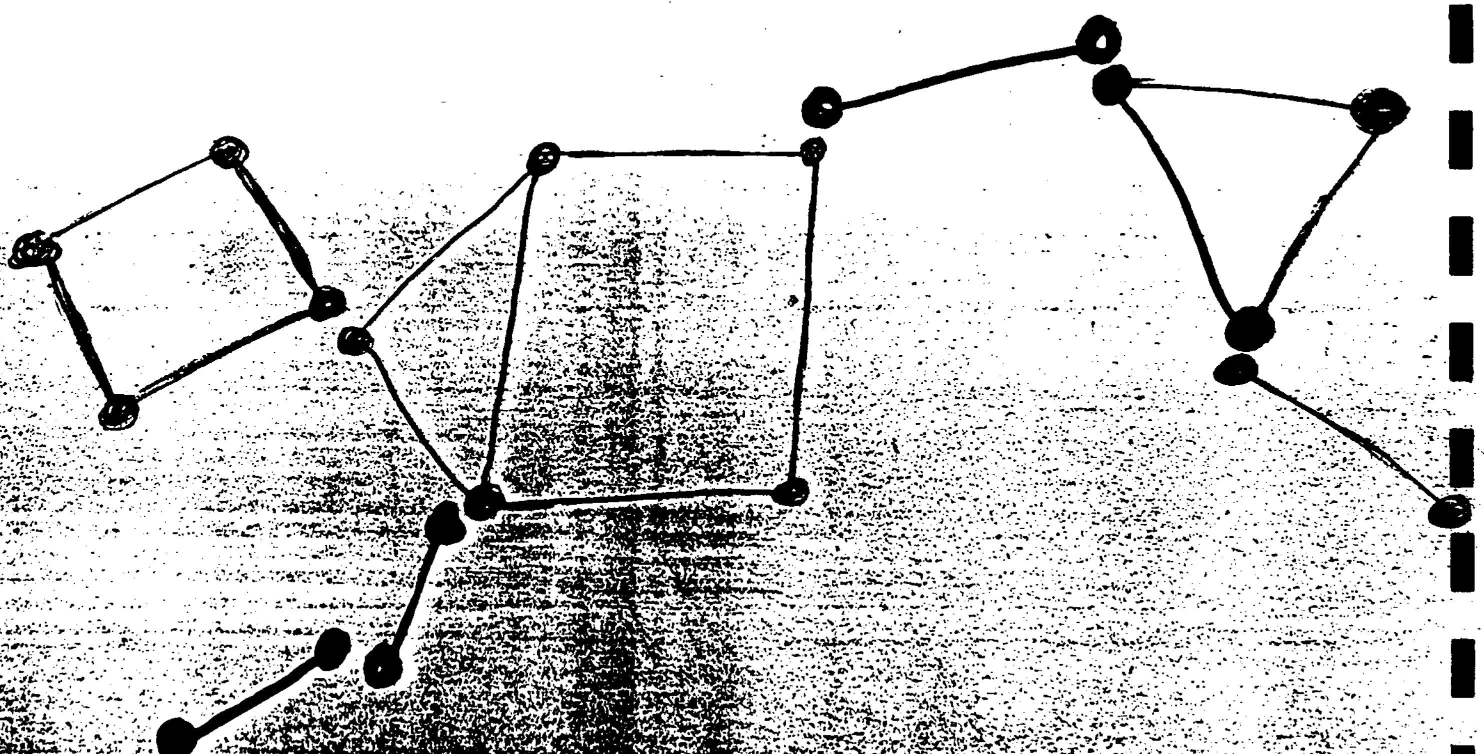
Theorem (Harary & Norman, 195[illegible]

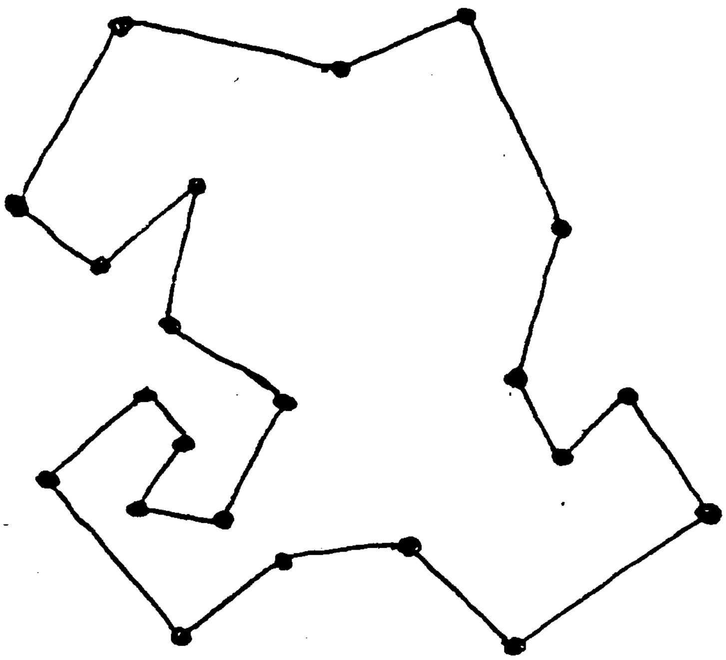
The center of any connected graph lies in a block.



A **block** is a maximal connected subgraph with no cutvertices

below: the blocks of the previous graph





A **geodesic center** of a simple polygon P is a point within

for which the distance to a farthest point is minimum

Theorem (**Asano & Toussaint**) The geodesic center of a simple polygon consists of a single point.

Geodesic Center Algorithms

Asano & Toussaint $n^3 \log n$

1989 - Pollack, Sharir, & Rote $n \log n$

<u>note</u>: Asano & Toussaint showed in fact that the geodesic center is located at either a vertex of the farthest point Voronoi diagram or at the midpoint of a geodesic diametral path.

<u>note</u>: Pollack et al. triangulate the polygon, do a binary search to find the triangle that contains the center point, and then use modified linear programming methods to find the point.

The **link distance** $\ell(x,y)$ between two points x and y within P is the smallest number of edges in a polygonal path (within P) joining x and y.

link radius, link diameter, link center

Lenhart, Pollack, Sack, Seidel, Sharir, Suri, Toussaint, Whitesides, & Yap

give an $O(n^2)$ algorithm to find the link center.

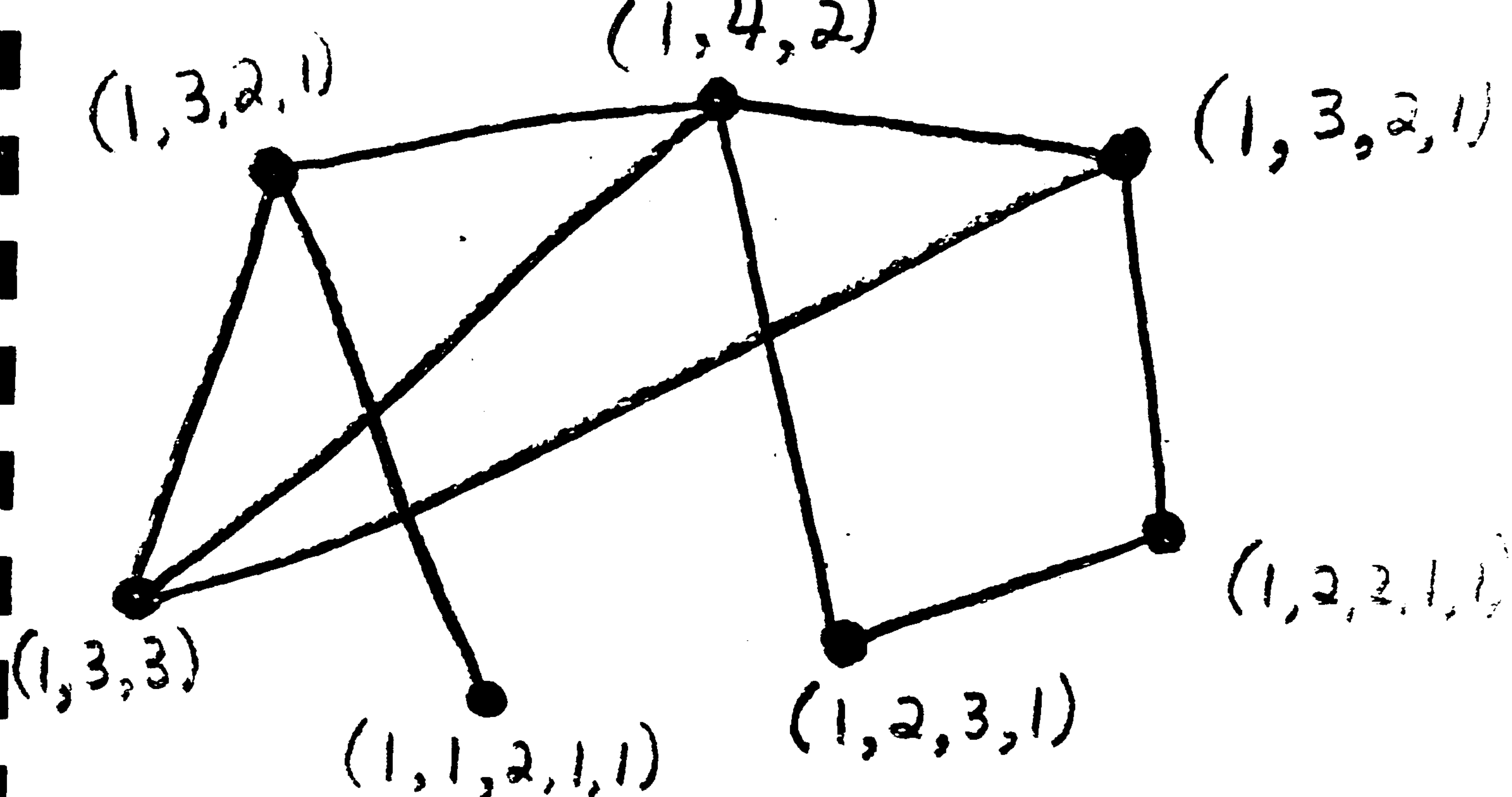
---

i^{th} visibility region (of v) the set of all points y with $\ell(v,y) = i$

A useful tool

For each vertex v in a connected graph G, let $d_i(v)$ be the number of vertices at distance i from v. The distance list at v is the sequence $(d_0, d_1, d_2, \ldots, d_n)$

The distance degree sequence for G is the sequence of distance lists for the vertices of G.



distance list

distance degree sequence

(1,1,2,1,1) ; (1,2,2,1,1) ; (1,2,3,1) ; $(1,3,2,1)^2$;
(1,3,3) ; (1,4,2)

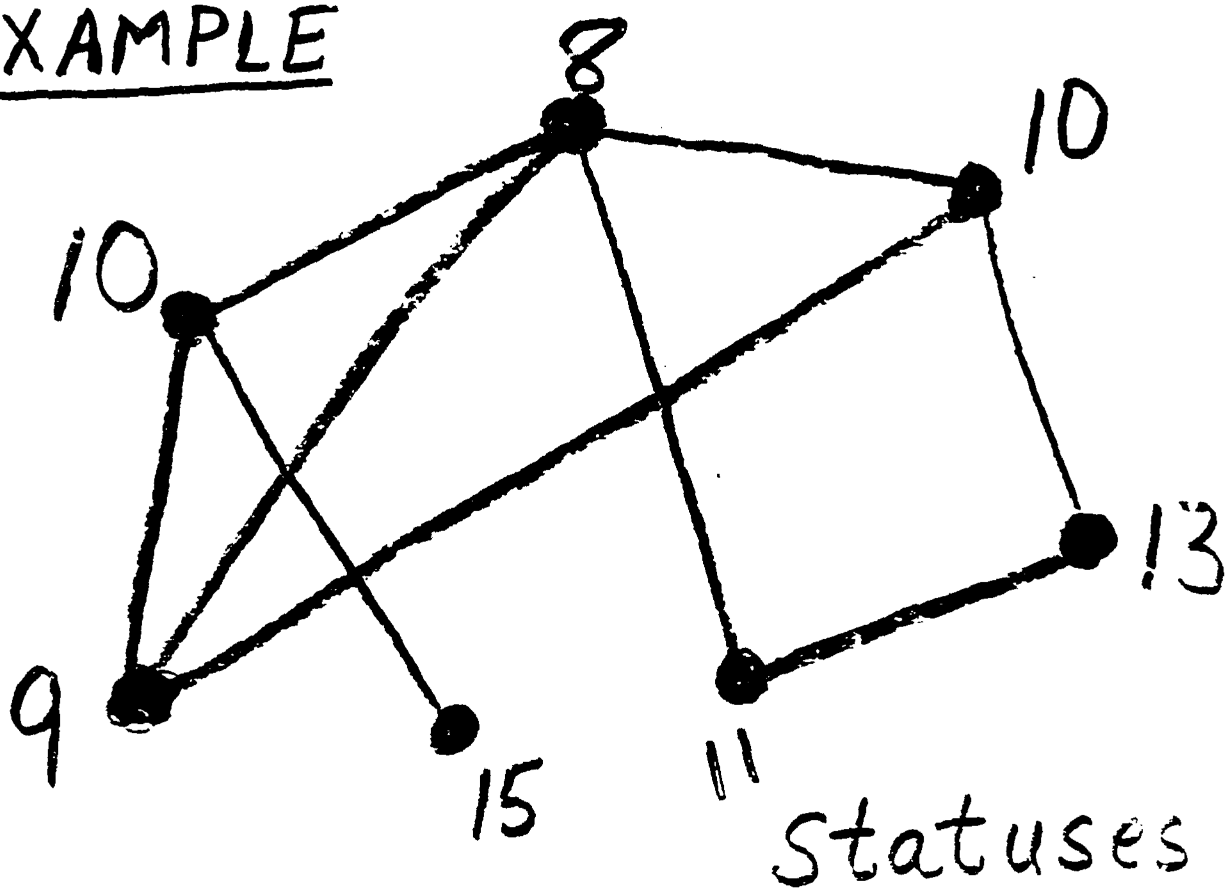
The *status* of a vertex v equals the sum of the distances from v to each other vertex in G

The *median* of a graph is the set of vertices that have minimum status

Notation $s(v)$

$M(G)$

EXAMPLE



Problem Examine the properties that a graph must have so that all of its statuses are distinct.

Theorem (Harary & Norman, 1953)
The center of any connected graph lies in a single block.

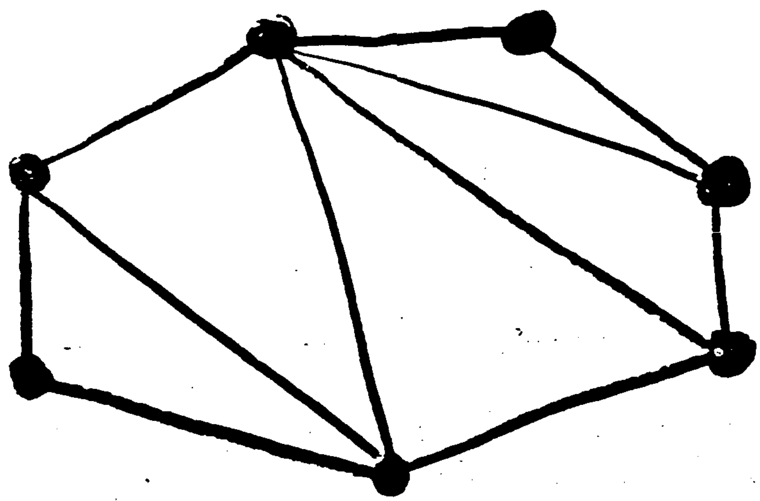
Structural Problems

Suppose G has some property (tree, chordal, outerplanar, hamiltonian). Can you determine the structure of $C(G)$. Is there a finite set S of graphs such that if G has property P, then $\langle C(G) \rangle \in S$?

A graph is outerplanar if it can be embedded in the plane so that all vertices are on the exterior face

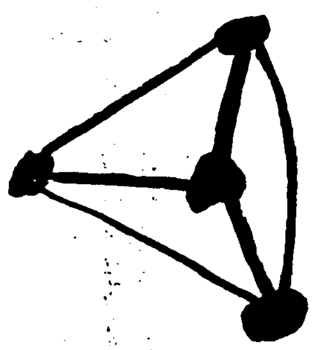
---

Example



is

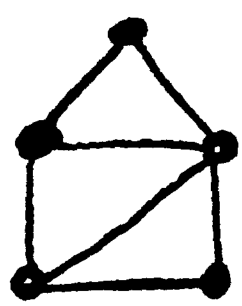
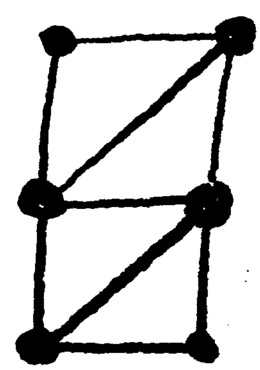
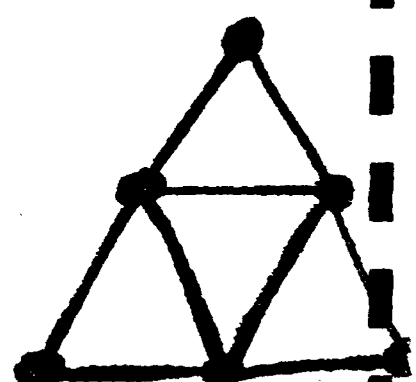
Example



is not

Theorem (Proskurowski, 1979)
If G is a maximal outerplanar graph, then $\langle C(G)\rangle$ is isomorphic to one of the seven graphs

K_1, K_2, K_3, $K_4 - e$,

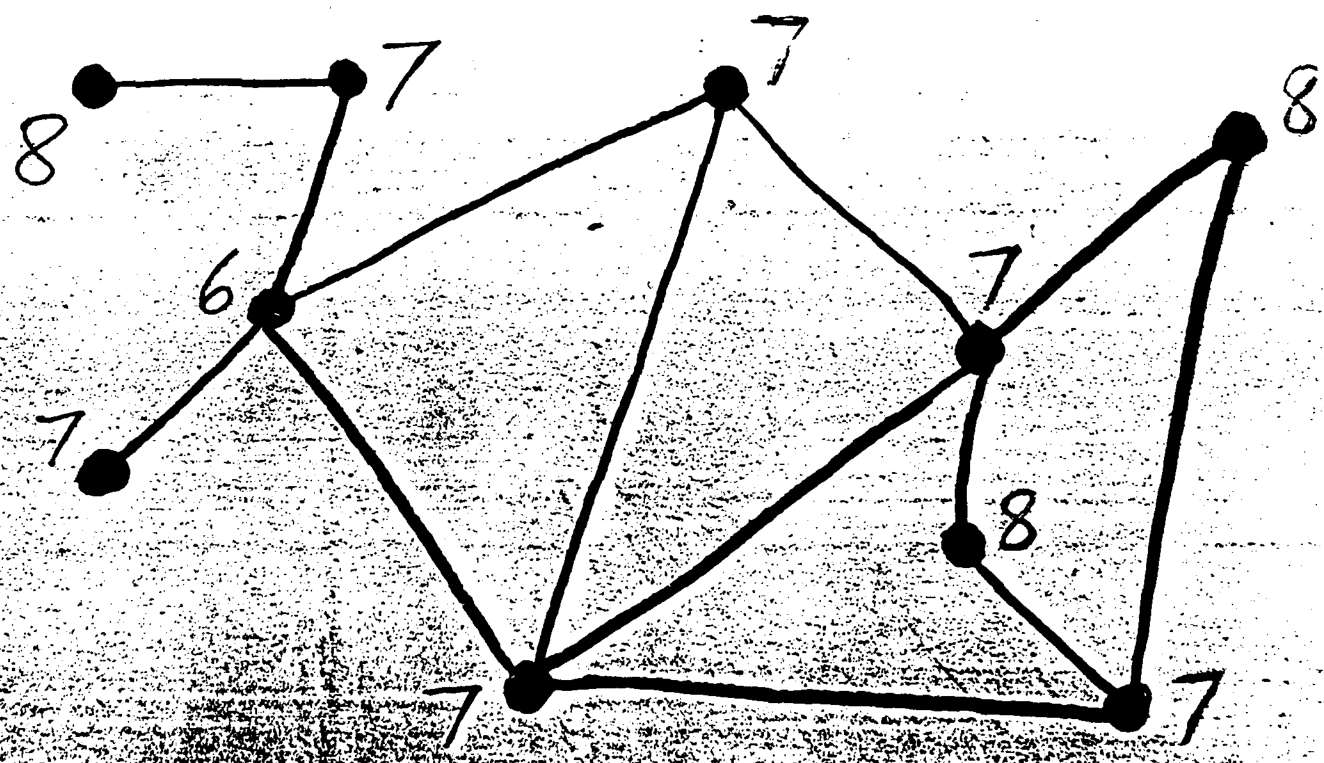

,

, or


(closed)

Question If H is a graph, then must H always be the central subgraph of some graph G?

Detours

A *detour path from u to v* is a path of maximum length joining u and v. The *detour number* $dn(v)$ of vertex v is the length of the longest detour path beginning at v.

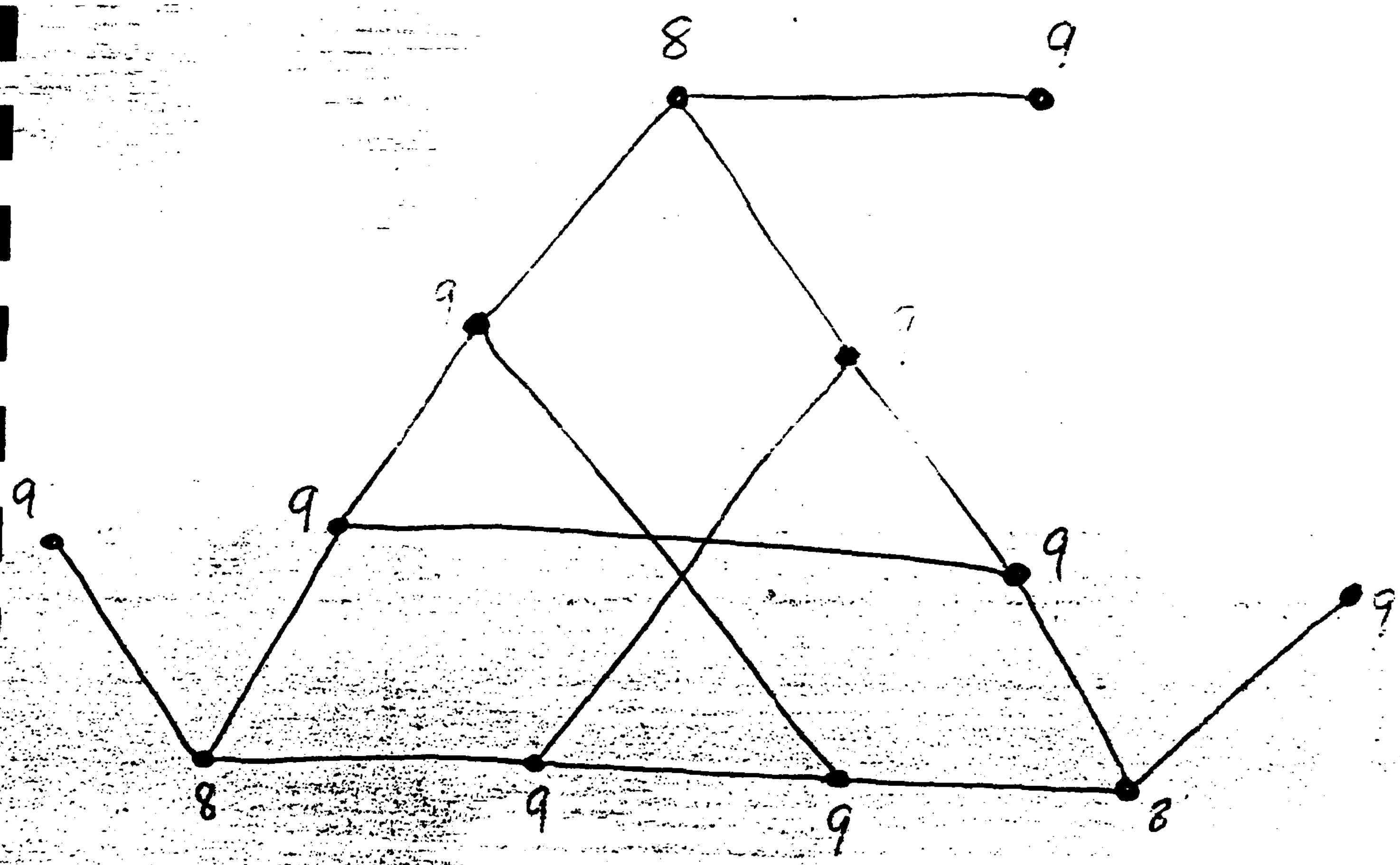


The **detour radius** of G is the minimum detour number of its vertices. The **detour center** of G is the set of vertices with minimum detour number.

Theorem (Kapoor, Kronk, & Lick, 1968) The detour center of a connected graph lies in a block.

Corollary The detour center of a tree consists of a single vertex or a pair of adjacent vertices.

a **detour in G** is a longest detour path in G



a graph with detours avoiding each vertex

(T. Zamfirescu, 1975)

Major References

1. F. Buckley and F. Harary, Distance in Graphs, Addison-Wesley, Reading, Mass. (1990).

2. T. Asano and G.T. Toussaint, Computing the geodesic center of a simple polygon, Discrete Algorithms and Complexity, Academic Press, Boston (1987) 65-79.

3. W. Lenhart, R. Pollack, J. Sack, R. Seidel, M. Sharir, S. Suri, G. Toussaint, S. Whitesides, and C. Yap, Computing the link center of a simple polygon, Disc. Comp. Geom. 3 (1988) 281-293.

ABSTRACT

Rectangle Proximity Graphs and Rectilinear Shortest Path Problems

D. T. Lee
Department of Electrical Engineering and Computer Science
Northwestern University
Evanston, Illinois 60201

We introduce the notion of rectangle proximity graph for a set of points in the plane. Given a set S of points in the plane, two points p and q are connected by an edge if the corresponding rectangle defined by p and q does not contain any points of *S in its interior or on the boundary*. The induced graph is called a rectangle proximity graph. It is shown that the rectangle proximity graph can be used to solve the rectilinear shortest path problem between two points in the presence of (rectilinear) obtacles. Although the proximity graph on a set of n points may have $O(n^2)$ edges, an appropriate representation, called shortest path preserving graph (SPPG), with O(nlogn) vertices and edges can be obtained so that the rectilinear shortest path problem can be solved in $O(nlog^{2n})$ time. An $O(nlog^{(3/2)n})$ time algorithm can also be obtained with a SPPG of size $O(nlog^{(3/2)n})$.

RECTANGLE PROXIMITY GRAPHS
and
RECTILINEAR SHORTEST PATH PROBLEMS

D. T. Lee

Department of Electrical Engineering
and Computer Science
Northwestern University

Division of Computer & Computation Research
National Science Foundation
Washington, D. C. 20550

Rectangle Proximity Graph – RPG

Graph $G = (V, E)$ in which V is a set of n points, $p_i, i = 1, 2, \ldots, n$ in the plane, and two points p_i, p_j are connected by an edge $(p_i, p_j) \in E$, **iff** the rectangle $R_{i,j}$ determined by these two points is *empty*, i.e., no other point in V lies in $R_{i,j}$.

Disk Proximity Graph – Gabriel Graph (GG)

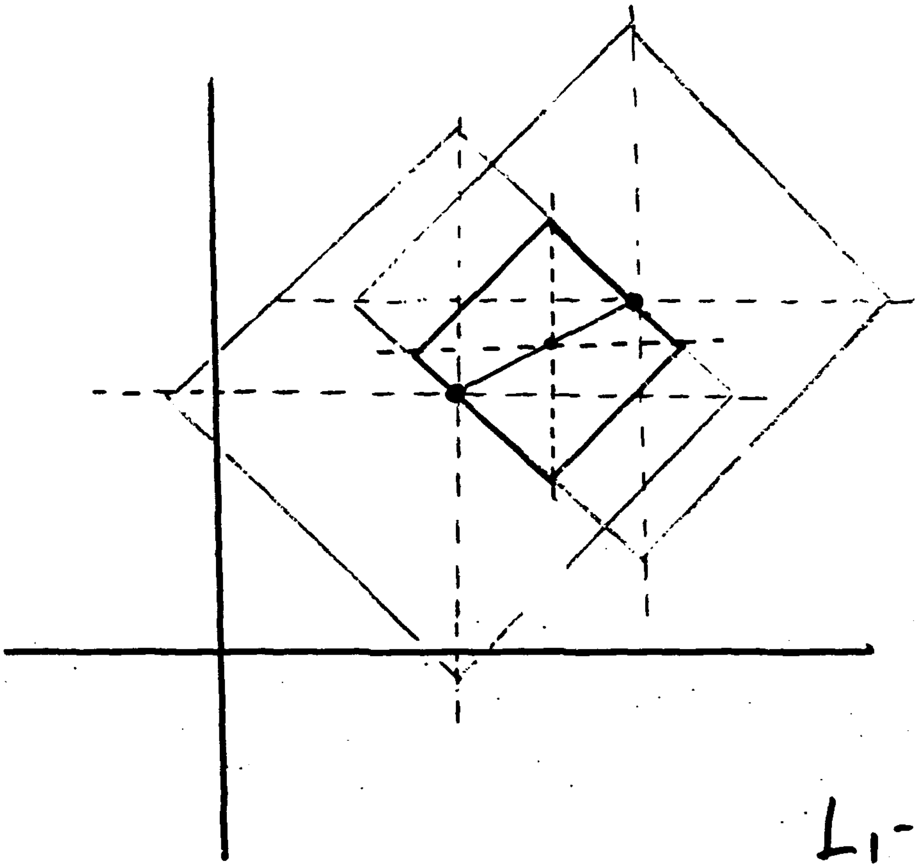
Graph $G = (V, E)$ in which V is a set of n points, $p_i, i = 1, 2, \ldots, n$ in the plane, and two points p_i, p_j are connected by an edge $(p_i, p_j) \in E$, **iff** the disk $D_{i,j}$ determined by these two points is *empty*, i.e., no other point in V lies in $D_{i,j}$.

Lune Proximity Graph – Relative Neighborhood Graph (RNG)

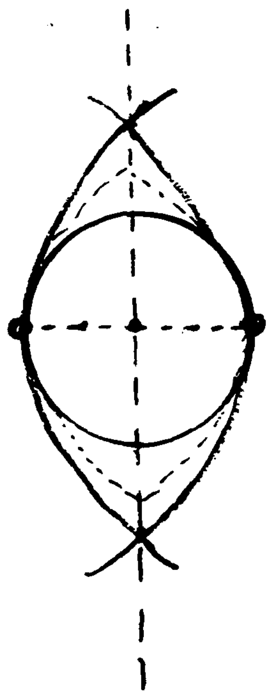
Graph $G = (V, E)$ in which V is a set of n points, $p_i, i = 1, 2, \ldots, n$. in the plane, and two points p_i, p_j are connected by an edge $(p_i, p_j) \in E$, **iff** the lune $L_{i,j}$ determined by these two points is *empty*, i.e., no other point in V lies in $L_{i,j}$.

Circle Proximity Graph – Delaunay Graph (DG)

Graph $G = (V, E)$ in which V is a set of n points, $p_i, i = 1, 2, \ldots, n$, in the plane, and two points p_i, p_j are connected by an edge $(p_i, p_j) \in E$, **iff** there exists a circle $\mathcal{K}_{i,j}$ passing through these two points is *empty*, i.e., no other point in V lies in $\mathcal{K}_{i,j}$.



L_1-metric



$\beta = 1$

$\beta = \frac{1}{2}$

$\beta = 2$

β-skeleton

Lune Proximity Graph – RNG $\subseteq$ Disk Proximity Graph – G(

Disk Proximity Graph – GG $\subseteq$ Circle Proximity Graph – DG

(Matula & Sokal)

(Shamos & Ho

Circle Proximity Graphs can be computed in $O(n \log n)$ time. (Lee & Schacht

Lune Proximity Graphs can be computed from DG in $O(n\alpha(n))$ time.

(Supowit) (Jaromczyk & Kowaluk

L_1-metric (or L_∞-metric) (O'Rourke, Lee)

RNG $\subseteq$ GG $\subseteq$ DG $\subseteq$ RPG

β-skeleton $G_\beta(V)$ $\beta \geq 1$ (Kirkpatrick & Radke)

$\mathcal{N}(p_i, p_j, \beta)$: intersection of circles of radius $\frac{\beta}{2} d(p_i, p_j)$ centered at points $(1-\frac{\beta}{2})p_i + (\frac{\beta}{2})p_j$ and $(\frac{\beta}{2})p_i + (1-\frac{\beta}{2})p_j$

$(p_i, p_j) \in E$ iff $\mathcal{N}(p_i, p_j, \beta)$ is empty

$\beta = 1$ Gabriel Graph

$\beta = 2$ RNG

β-skeleton $(1 \leq \beta \leq 2)$ can be computed from DG in $O(n)$ time

(Jaromczyk, Kowaluk & Yao)

Algorithm RPG for a set of points

Complexity $O(n \log n) + \mathcal{K}$, $\mathcal{K}$ is the output size.
Divide-and-Conquer Approach

Step 1 Divide S into two subsets S_1 and S_2 by a vertical line $\mathcal{V}$.

Step 2 Recursively build the **RPG's** for S_1 and for S_2.

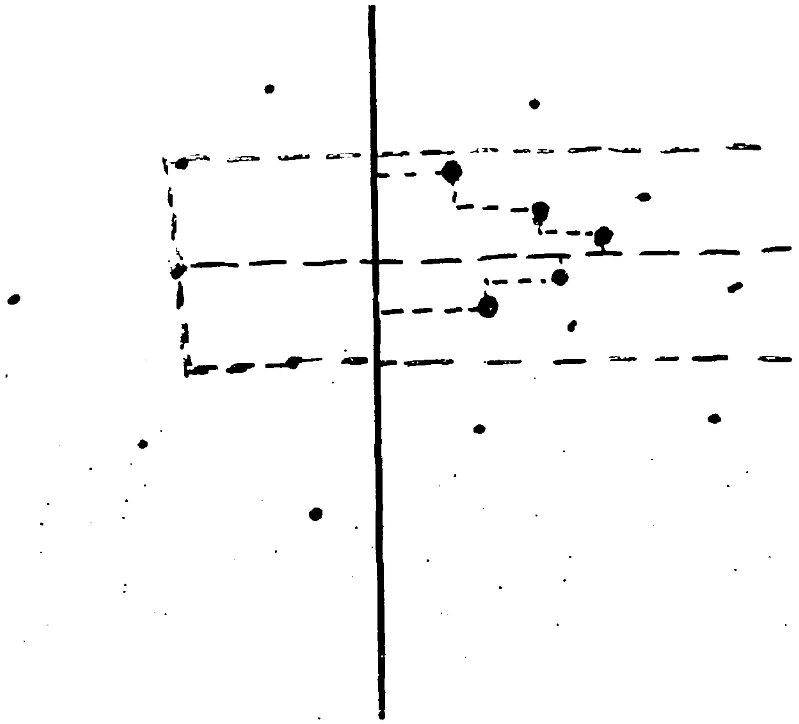
Step 3 Construct edges connecting points in S_1 and in S_2 as follows.

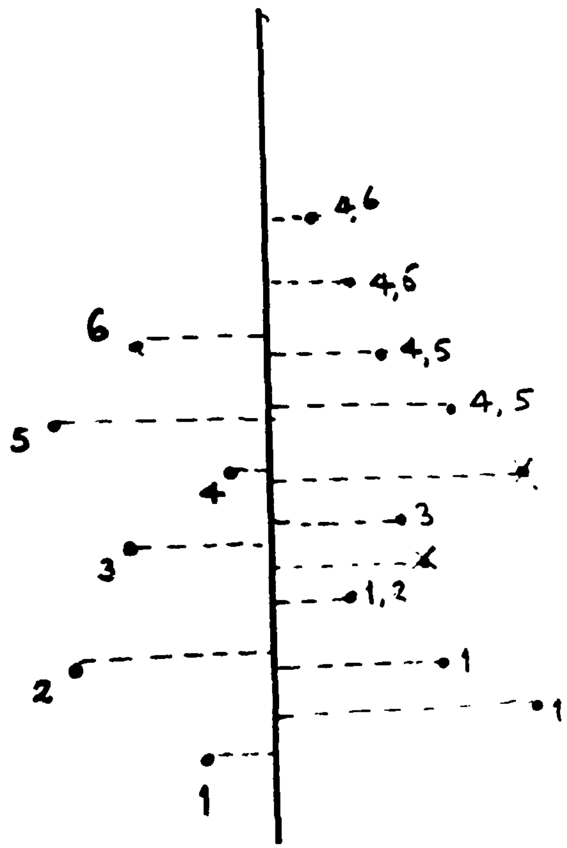
- **Step 3.1** Scan the points in S from bottom up, and maintain a 'staircase' for each point in S_1 and in S_2.
  Let $\mathcal{S}_1$ and $\mathcal{S}_2$ denote the sets of staircases for points in S_1 and in S_2 respectively.
- **Step 3.2** If the next point p_i is in S_1, consult the staircases in $\mathcal{S}_2$ and **decide** if an edge is to be introduced for p_i and points in S_2.
- **Step 3.3** If the next point p_i is in S_2, consult the staircases in $\mathcal{S}_1$ and **decide** if an edge is to be introduced for p_i and points in S_1.

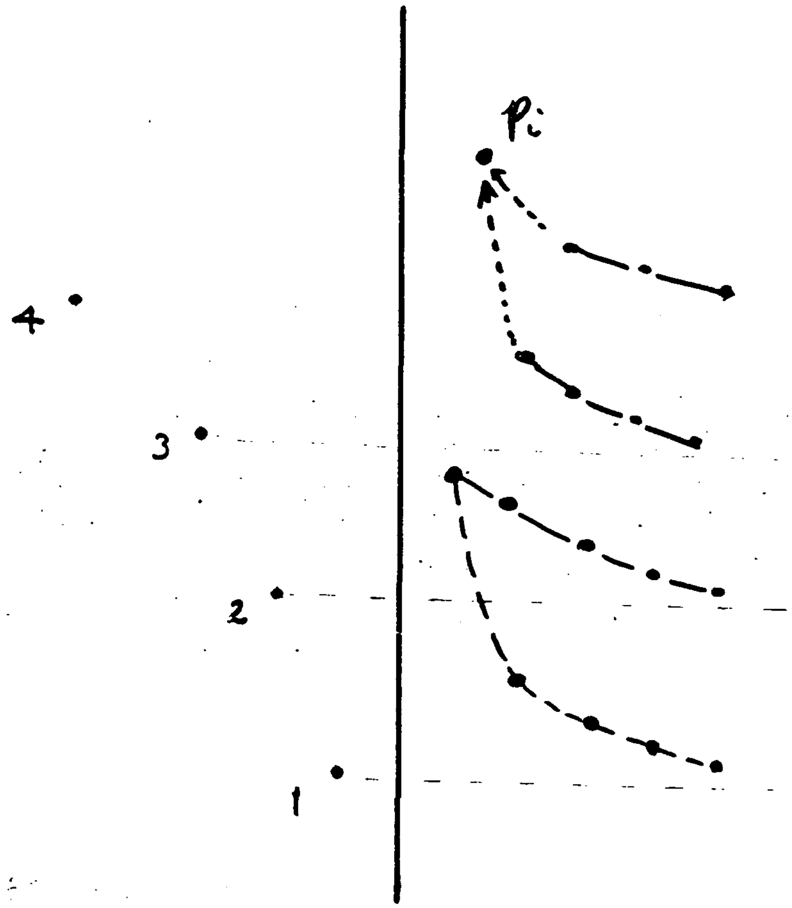
Note: Only the **topmost** point of each staircase is needed.
~~**Binary Search** is performed in **Steps 3.2 and 3.3**~~

$$T(n) = 2T(n/2) + O(n) + \underline{\underline{K}}$$

$$O(n \log n) + K$$




4,6
4,6
6
4,5
4,5
5
4
3
3
1,2
2
1
1
1



left staircase p_{l_4} p_{l_3} p_{l_2} p_{l_1} p_{r_4}

right staircase (topmost elements) p_{r_3} p_{r_2}, p_{r_1}

P_l : update left staircase

P_r : update right staircase & "output"

Rectilinear Shortest Path Problem
Given n isothetic rectilinear obstacles $R_1, R_2, \ldots, R_n$, each with a positive weight $R_i.w, i = 1, 2, \ldots, n$, and two distinguished points s and t, in the plane, find a shortest rectilinear path connecting s and t.

Definition 1

Let Π_{st} denote a rectilinear path connecting s and t.
Π_{st} is denoted as: $q_1, p_1, q_2, p_2, q_3, p_3, \ldots q_k, p_k$, where q_i is a subpath outside any obstacle, and p_i is a path completely within R_i.
q_1 or p_n may be empty.

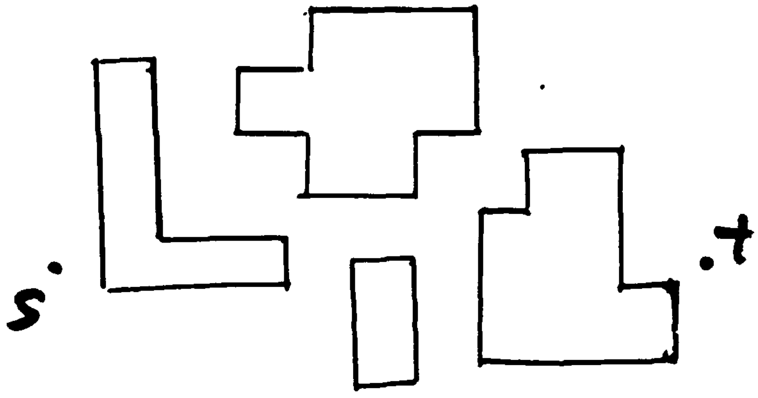
Weighted length:

$$dw(\Pi_{st}) = \sum_{i=1}^{k}(|q_i| + |p_i|) + \sum_{i=1}^{k}(R_i.w * |p_i|).$$

Note: *Collision Free Path:* $R_i.w = \infty$ for all i.
$p_i = \phi, i = 1, 2, \ldots, k.$

Optimal Path: Π^*_{st}

Path Length: dw^*_{st}



$V \equiv$ set of vertices of obstacles $R_i \cup \{s,t\}$.

$I \equiv$ set of internal projections

$\{p_r, p_l, p_u, p_d | p \text{ a concave vertex}\}$

$\cup\{q_r, q_l, q_u, q_d | q \in \{s,t\}, q \in R_i\}$

$\overline{V} = V \cup I$.

Definition 2

Point p **1-dominates** point q iff $p.x \geq q.x$ and $p.y \geq q.y$.
Point p **1-directly-dominates** point q iff there exists no point r such that p 1-dominates r and r 1-dominates q.
i-(directly)-dominating relations are defined similarly $i = 2, 3, 4$.

Definition 3

A vertex p of obstacle Q is **1-directed** iff the boundary edges incident on p are in the $+X$ and $+Y$ directions, abbreviated as $(+X, +Y)$ directions. Similarly, we define 2, 3 and 4-directed vertices if the boundary edges are in $(+Y, -X)$, $(-X, -Y)$ and $(+X, -Y)$ directions respectively.

Definition 4

For $p \in \overline{V}$, $S_i(p) = \{q | q \in \overline{V}$ and

q i-directly dominates $p \in \overline{V}\}$.

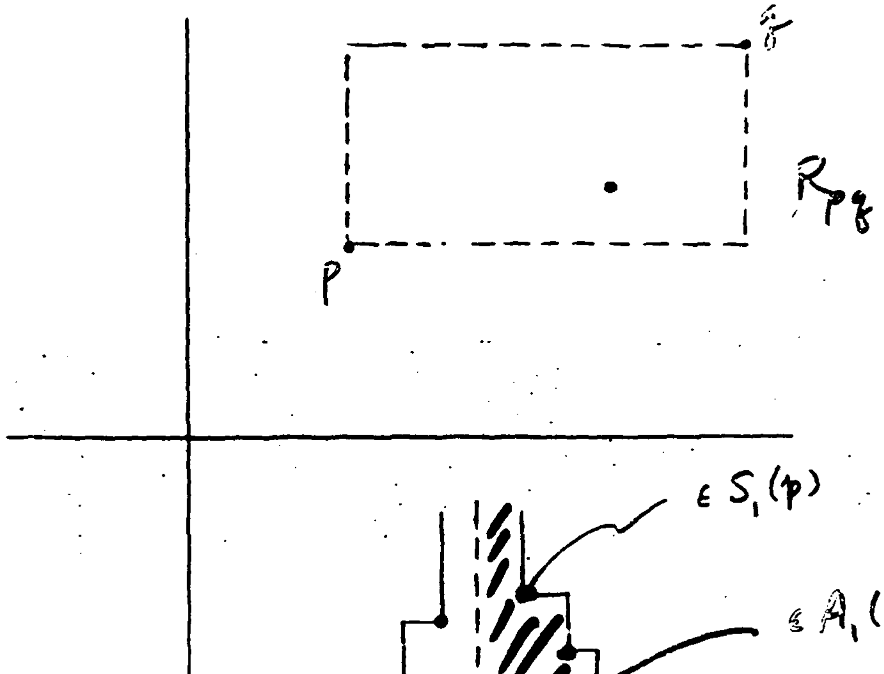
$S(p) = \cup_i S_i(p)$.

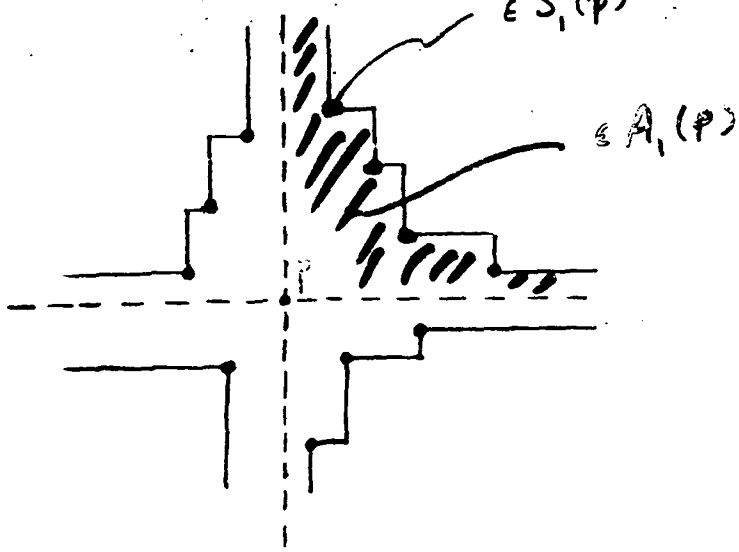
$A_i(p) = \{q | q$ i-dominates p, and

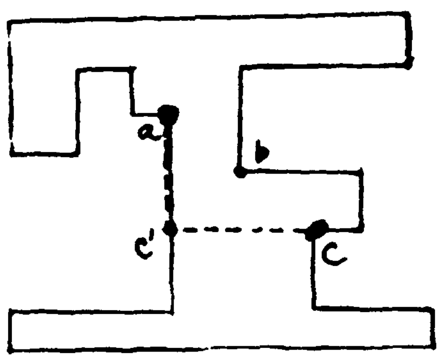
q does not i-dominate $r, r \in S_i(p)\}$.

$A(p) = \cup_i A_i(p)$.

i-staircase $SC_i(p)$ = boundary of region defined by $A_i(p)$.


q
R_pq
p


∈ S_1(p)
∈ A_1(p)
p


a
b
c'
c

Lemma 1

$A(p) \cap \overline{V} = \{p\}$, i.e., $A(p)$ contains no point of $\overline{V}$ except p itself.

Lemma 2

$A_i(p), i = 1, 2, 3, 4$ satisfies one of the following properties:

1: $A_i(p)$ is totally outside any obstacle,

2: $A_i(p)$ is totally within an obstacle,

3: $A_i(p)$ contains only vertical strips, or

4: $A_i(p)$ contains only horizontal strips.

Lemma 3

Let p be a concave vertex of obstacle Q with internal projections p_r and p_u. Then $A_i(p), i = 1, 2, 4$ are all within Q and both $A_2(p)$ and $A_4(p)$ are rectangles.

Lemma 4

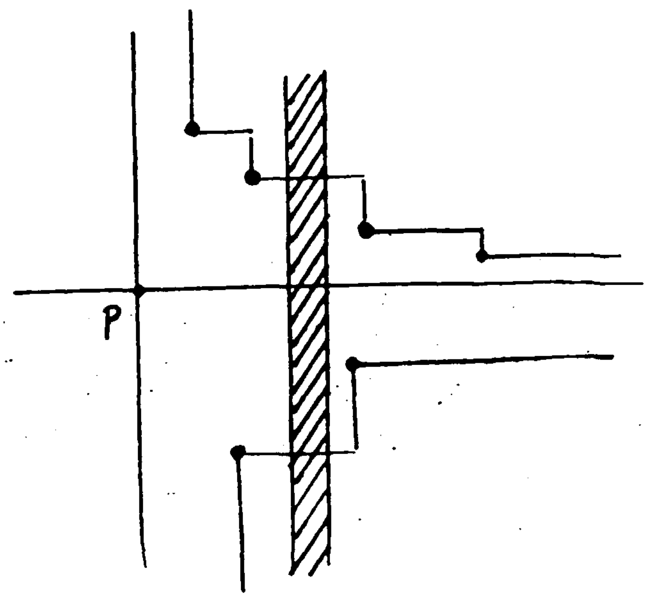
Let p be an i-directed convex vertex of Q.
$A_i(p)$ is a rectangle within Q. Other symmetric cases hold.

Lemma 5

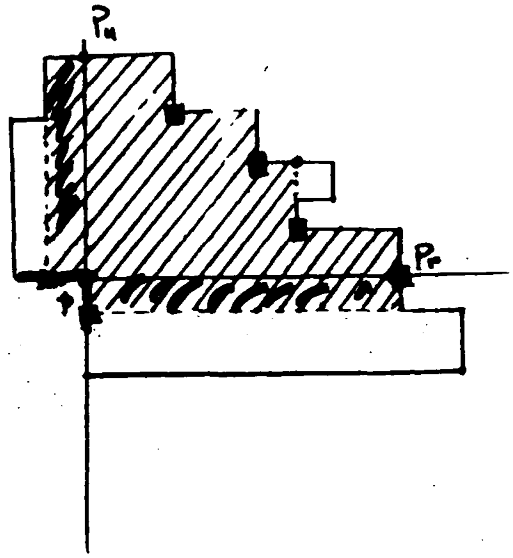
Consider an internal projection, say p_d, of a concave vertex p of Q. Both $A_1(p_d)$ and $A_2(p_d)$ are rectangles within Q. Other similar cases hold.

Lemma 6

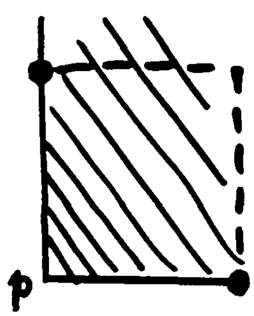
If s (resp. t) lies in Q, $A(s)$ (resp. $A(t)$) lies in Q.



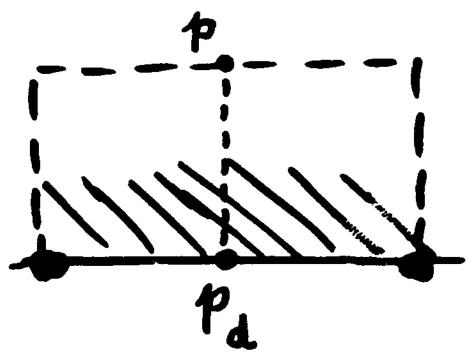
Lemma 2



Lemma 3



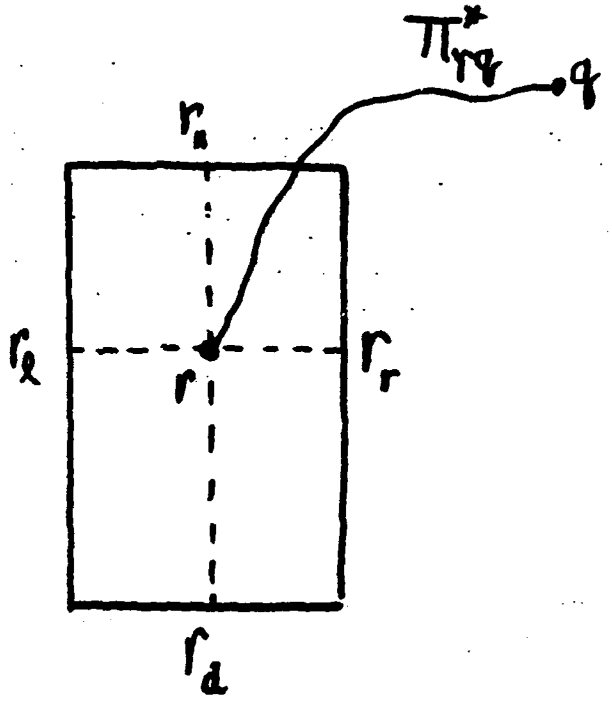
Lemma 4



Lemma 5

Lemma 7

Consider a rectangular area R in obstacle Q, and let $r \in Q$. Then fo[r] any point $q \notin int(R)$, there exists a shortest path Π_{rq} that passes through one of the projections r_u, r_d, r_l, or r_r on $bd(R)$.



Theorem 1

For any two points $u, w \in V$, $w \notin A(u)$, there exists a shortest path, Π^*_{uw}, that passes through at least one point of $S(u)$.
Proof: Consider only the cases in the 1st quadrant with u being the origin.
Let p and q be two consecutive points in $S_1(u)$. Assume that $\Pi^*_{uw} = \Pi^*_{ur}||\Pi^*_{rw}$, where $r \in SC_1(u)$ lies on horizontal part between p and q, and Π^*_{ur} totally lies in $A_1(u)$.
Let p' be the left projection of p on the Y-axis.

Case 1 $A_1(u)$ outside of any obstacle.

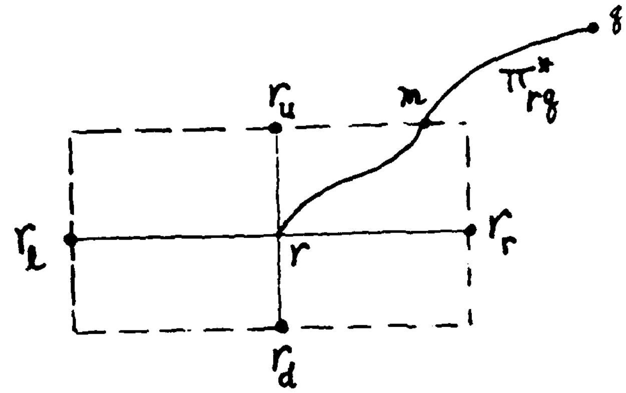
Case 2 $A_1(u)$ contains only vertical strips.

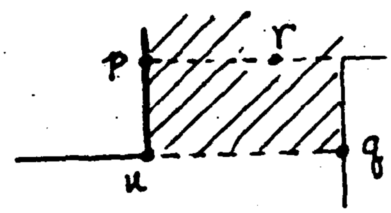
Case 3 $A_1(u)$ contains only horizontal strips.

Case 4 $A_1(u)$ totally lies in an obtacle Q.
This case is more complicated.

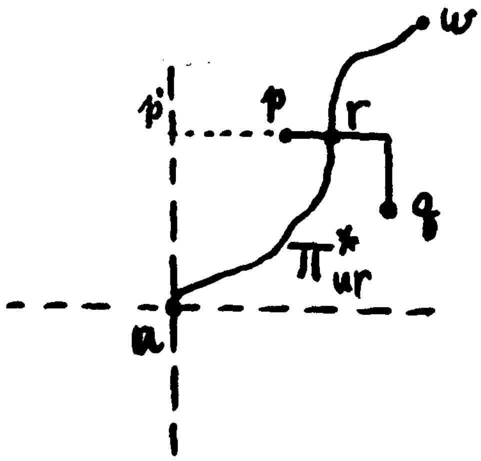
Case 4.1 u is a concave vertex.
case 4.1.1 u is 3-directed.
$\Pi^*_{ur} = \overline{up'}||\overline{p'r}$.

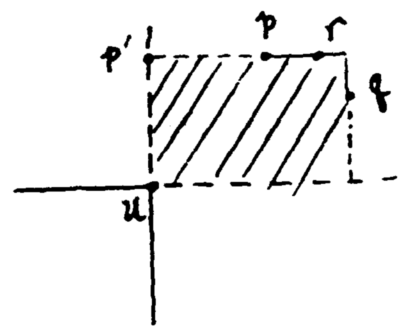
case 4.1.2 u is 2-directed.
$A_1(u)$ is a rectangle R_{pq} and $p' = p$.
$\Pi^*_{ur} = \overline{up}||\overline{pr}$
case 4.1.3 u is 4-directed.
If $\overline{pr}$ lies on $bd(Q)$, done.
Since $A_1(u)$ is a rectagle $(u, q, m, l) \in Q$, from Lemma 7, Π^*_{rw} must pass though one of the following:
(a) Through r_d: Impossible.



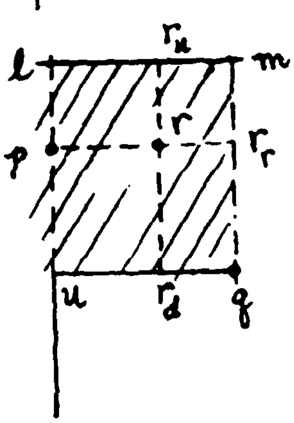


u 2-directed





3-directed



4-directed

(b) Through r_r: $\Pi^*_{ur}||\Pi^*_{rw} = \overline{uq}||\overline{qr_r}||\Pi^*_{r_r w}$

(c) Through r_u: $\Pi^*_{ur}||\Pi^*_{rw} = \overline{ul}||overlinelr_u||\Pi^*_{r_u w}$

(d) Through $r_l = p$: Done.

Case 4.2 u is convex.

Case 4.3 $u \in I$. Similar to **Case 4.1.3**.

Case 4.4 $u \in \{s, t\}$. $A(u) \in Q$.

Algorithm RSP– Graph-Theoretic Approach

Step 1 Compute $\overline{V} = V \cup I$.

Step 2 Compute for each $v \in \overline{V}$ its staircase $SC(v)$.

Step 3 Compute the **Graph** $G = (\overline{V}, E)$, where $(v, u) \in E$ if $u \in SC(v)$; assign **weight** to each edge.

Step 4 Apply Fredman/Tarjan's algorithm on G.

In the worst case E has $O(n^2)$ edges and it needs $\Omega(n^2)$ time to construct G.

$O(|E| + |V| \log |V|)$

$O(n \log^2 n)$ time $O(n \log n)$ space

$O(n \log^{3/2} n)$ time & Space

Modifications to G – Adding Steiner Points

Definition 5

Points p and q are said to be *visiable* from each other if the segment $\overline{pq}$ is either totally outside all the obstacles or totally inside some obstacle. A line L is said to be *visible* from a point p if and only if the perpendicular projection p' of p on L is visible from p.

Lemma 8

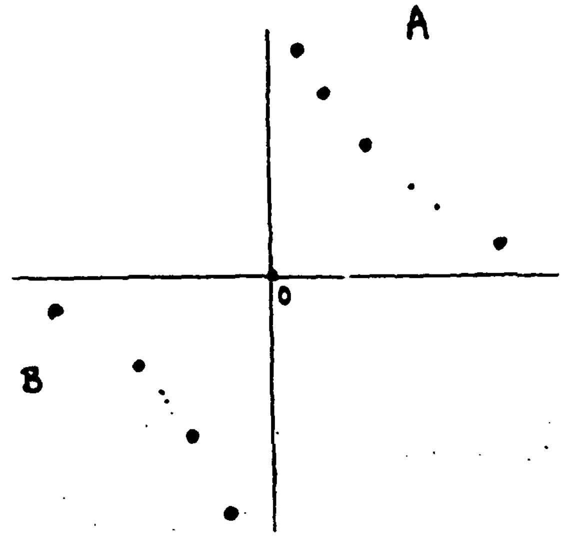
The path Π^*_{pq} represented by an edge (p, q) for any $p \in V$ and $q \in S(p)$, can be obtained by one of the following.

case 1 p and q are both visible from a vertical line $\mathcal{V}$ separating them.
$\Pi^*_{pq} = \overline{pp'} || \overline{p'q'} || \overline{q'q}$, where p' and q' are projections of p and q on $\mathcal{V}$.

case 2 p and q are both visible from a horizontal line $\mathcal{H}$ separating them.
$\Pi^*_{pq} = \overline{pp'} || \overline{p'q'} || \overline{q'q}$, where p' and q' are projections of p and q on $\mathcal{H}$

case 3 $p.x = q.x$ or $p.y = q.y$.
$\Pi^*_{pq} = \overline{pq}$.

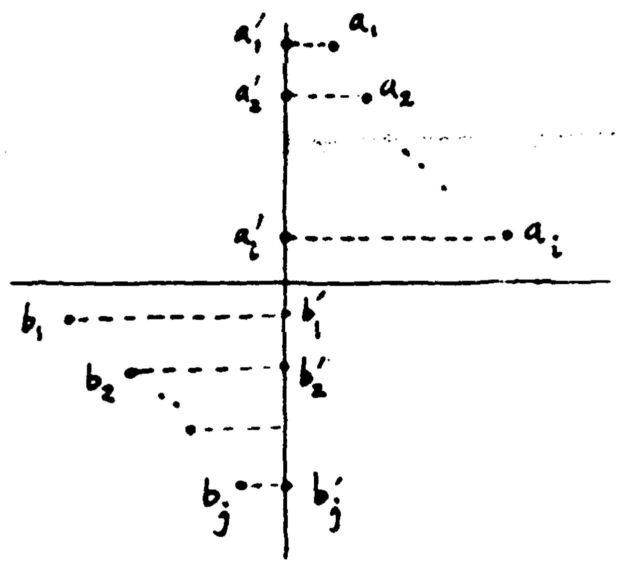
The projection points are Steiner points.



$$G = (A \cup B, E)$$

$$|E| = O(|A| * |B|)$$

$$|ab| = |ao| + |ob|$$



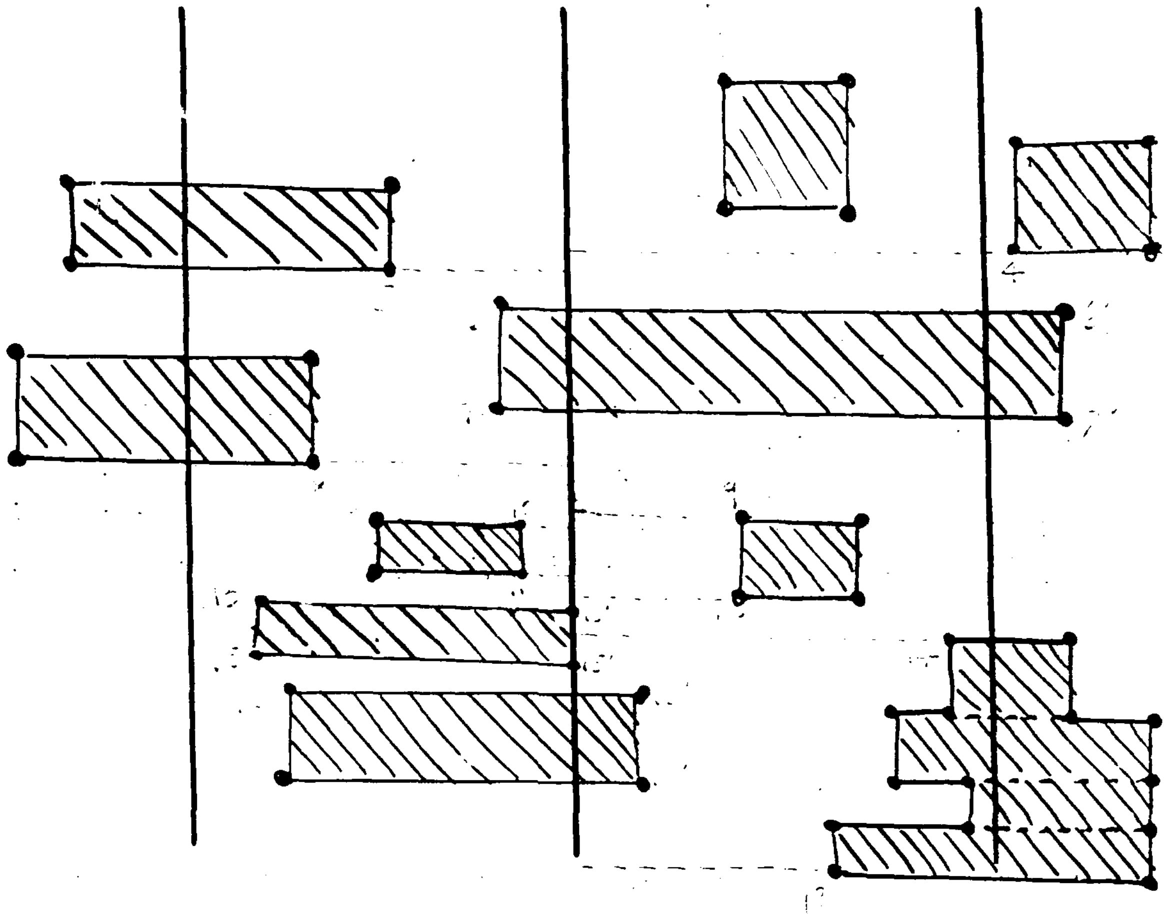
$$G' = (A \cup B \cup I, E')$$

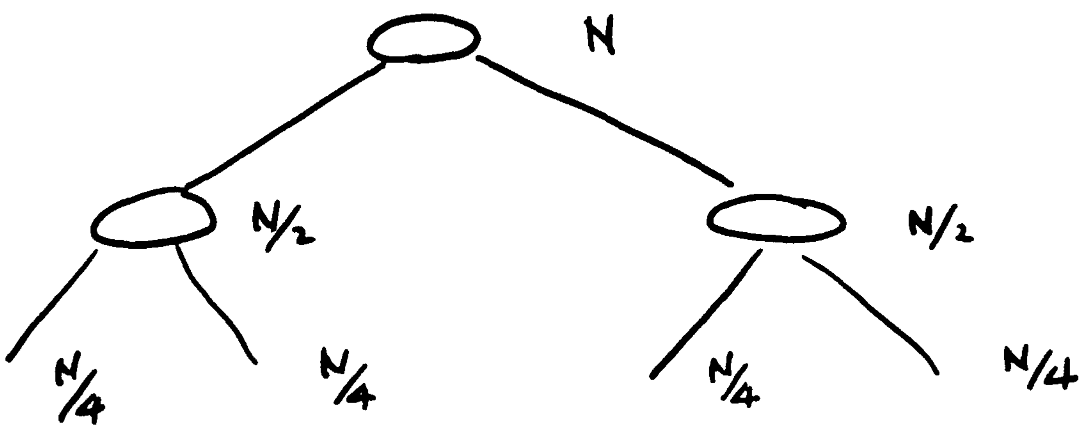
$$I = \{a_1', a_2', \cdots, a_i', b_1', \cdots, b_j'\}$$

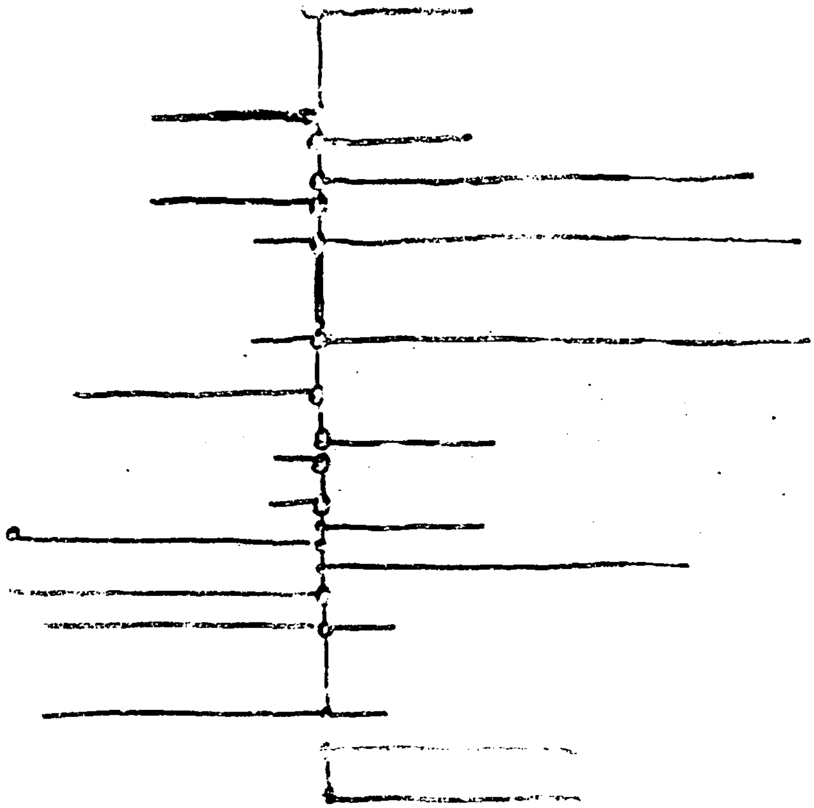
$$E' = \{(a, a'), (b, b'), (a_1', a_2') \cdots (a_i', b_1'), \cdots (b_{j-1}', b_j')\}$$

$$|E'| = O(|A| + |B|)$$

I: Steiner Points.




N
N/2
N/2
N/4
N/4
N/4
N/4



$E = O(N)$

Construction of Weighted Graph G'

- Draw a vertical "cut" line $\mathcal{V}$ through the median of the X-coordinates of all the points in $\overline{V}$.
  A horizontal cut line is drawn similarly. (Only vertical case is given.)
- Let $\overline{V_L}$ and $\overline{V_R}$ denote the vertices of $\overline{V}$ that lie to the left and right of $\mathcal{V}$ respectively, and $\overline{V_M}$ the vertices of $\overline{V}$ that lie on $\mathcal{V}$.
  For each point $p \in \overline{V_L} \cup \overline{V_R}$, if p is visible from $\mathcal{V}$, we create a Steiner point p' and the edge $(p, p') \in E$ with weight equal to $|\overline{p, p'}|$.
  We create for every two consecutive points on $\mathcal{V}$ an edge in E with weight computed using a plane sweep method.
- Recursively do the same thing on the sets $\overline{V_L}$ and $\overline{V_R}$ respectively.

Compute the weights of edges connecting two consecutive points on every cut line $\mathcal{V}_i, i = 1, 2, \ldots N$, by plane sweep.

step 1 Preprocess the obstacles by partitioning each of them into rectangles by introducing horizontal segments.

step 2 Sweep downward the ordinate of of those Steiner points and horizontal edges of the rectangles in the order of l, p and u.
Consider, the Steiner points that are on some cut line $\mathcal{V}_i$, with which two attributes $\mathcal{V}_i.w$ and $\mathcal{V}_i.s$ are associated.
$\mathcal{V}_i.w \equiv$ the accumulated weight and
$\mathcal{V}_i.s \equiv$ the last swept Steiner point on $\mathcal{V}_i$.

step 2.1 When the sweep line reaches an upper edge (u) of a rectangle, do nothing.

step 2.2 When it reaches a lower edge (l) of a rectangle, add the weight of that rectangle (i.e., the product of the weight and height of the rectangle) into $\mathcal{V}_i.w$.

step 2.3 When a Steiner point (p) is reached, we can calculate the weighted distance between it and the last one (recorded in $\mathcal{V}_i.s$); set $\mathcal{V}_i.s$ to the current Steiner point and reset $\mathcal{V}_i.w$.

Lemma 9

The graph $G = (V_{G'}, E_{G'})$ generated in **algorithm RSP** has $O(n \log n)$ vertices and edges.

Theorem 2

Algorithm RSP runs in $O(n \log^2 n)$ time and $O(n \log n)$ space.

Bottleneck of **Algorithm RSP**: Size of graph is $O(n \log n)$. Consequently, steps 3 and 4 run in $O(n \log^2 n)$ time.

Alternative Approach – Trading #vertices with #edges

Idea: Partition the points in each recursive step into **strips** such that each strip contains $\sqrt{\log n}$ points.

Need much fewer Steiner points.
For points in the same strip, we create edges between all pairs.
As a result, # vertices is $O(n \log^{1/2} n)$, and # edges is $O(n \log^{3/2} n)$.

Algorithm RSP'– $O(n \log^{3/2} n)$ Time and Space

This algorithm is the same as RSP except that we replace step 3 by the following:

3. (Construct Steiner points and add extra edges)

3.1 Partition the points into horizontal strips such that each strip contains $O(\sqrt{\log n})$ points, where n is the total number of vertices.
Divide the points by a line $\mathcal{V}$ with $\mathcal{V}.x$ equal to the median of all the X-coordinates of the points.
In each strip, we keep only the two extreme Steiner points, $\int_u, \int_d$.
Insert edges between the two Steiner points and the corresponding points.

3.2 (Construct edges in each strip)
Let H denote the set of points horizontally visible from $\mathcal{V}$ and $\int_u, \int_d$.
Let $H.l$ denote the set of points in H that are on $\mathcal{V}$ and to the left of $\mathcal{V}$, and let $H.r$ denote the set of points in H that are on $\mathcal{V}$ and to the right of $\mathcal{V}$.

3.2.1 Calculate the weighted path length between every two consecutive potential Steiner points (projections of points in H) on $\mathcal{V}$.

3.2.2 Create a table storing the weighted distance between every pair of points in H using results obtained in **step 3.2.1**.

3.2.3 With the table we construct a new table storing the weighted distance between each pair of points one in the $H.l$ and the other in $H.r$.

3.3 Do the same thing recursively to the set of points to the left and to the right of $\mathcal{V}$.

Lemma 10

The graph $G = (V_{G'}, E_{G'})$ generated in **Algorithm RSP'** has $O(n\sqrt{\log n})$ vertices and $O(n \log^{3/2} n)$ edges. **Proof:**
There are $O(\frac{n}{\sqrt{\log n}})$ strips, each containing $O(\sqrt{\log n})$ points. The total number of strips is $O(n\sqrt{\log n})$.
Only 2 Steiner points per strip are added to $V_{G'}$.

The number of edges connecting Steiner points and the points from which they are projected is $O(n\sqrt{\log n})$.
The number of edges that are constructed for the points in each strip is $O(\log n)$.
Thus the total number of edges in $E_{G'}$ is $O(n \log^{3/2} n)$.

Theorem 3

The algorithm **RSP'** computes the shortest path from the point s to the point t in $O(n \log^{3/2} n)$ time and space.

Open Problems

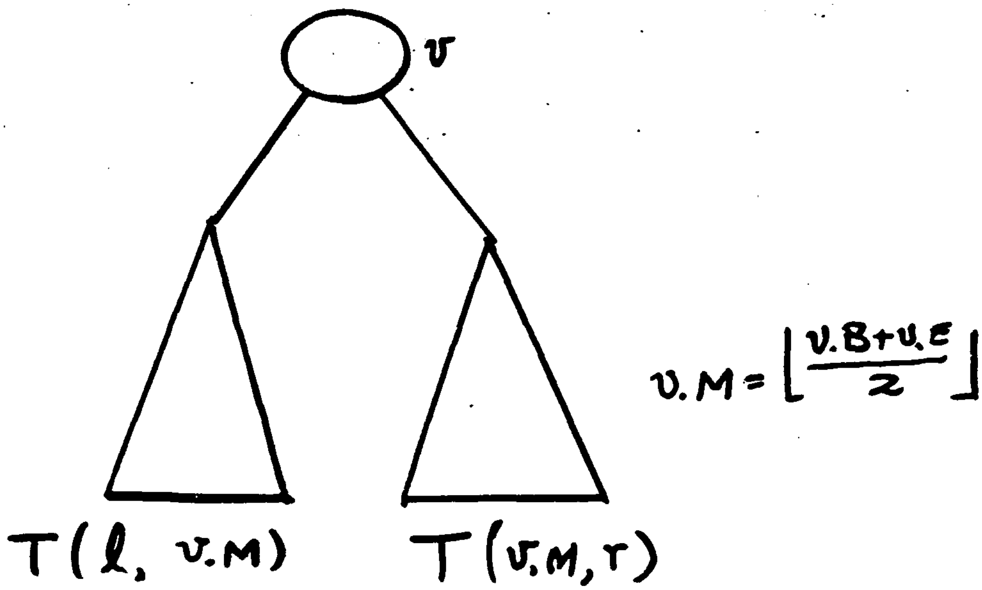
- Can one compute the shortest path in $O(n \log n)$ time and $O(n)$ space?
- What if the obstacles are of arbitrary shape?
- ~~Can **RPG** be computed in $O(n \log n)$ time $+ \mathcal{K}$?~~

* RPG in d-dimensions

* Proximity graphs in higher dimensions.

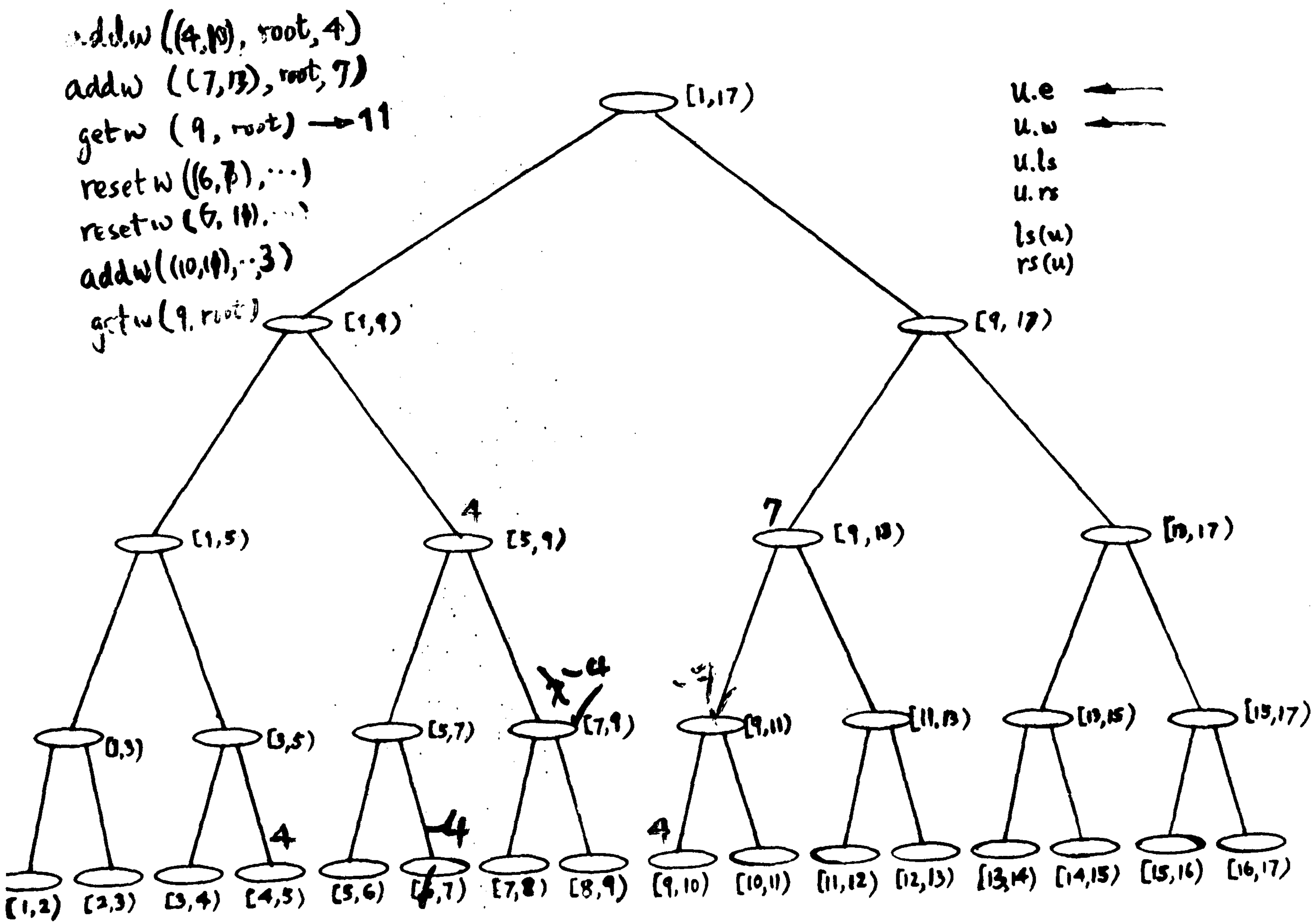
Weighted Segment Trees

Segment trees for a set of intervals on the real line with n endpoints normalized to integers in the range $[1, n+1)$.
Given integers $l, r, l < r$, segment tree $T(l, r)$ is recursively built as follows:
It consists of a root v, with attributes $v.B = l$ and $v.E = r$, and if $r - l > 1$, of a left subtree $T(l, \lfloor \frac{v.B+v.E}{2} \rfloor)$ and a right subtree $T(\lfloor \frac{v.B+v.E}{2} \rfloor, r)$.



standard intervals:
$[v.B, v.E)$ for each node v.
elementary intervals:
$[l, l+1), [l+1, l+2), \ldots, [n, n+1)$ – leaf nodes

For $r - l > 3$, an arbitrary interval $[b, e)$, $b < e$, is partitioned into a collection, called **canonical covering** of (b, e), of at most $\lceil \log_2(r-l) \rceil + \lfloor \log_2(r-l) \rfloor - 2$ standard intervals of $T(l, r)$.


addw ((4,10), root, 4)
addw ((7,13), root, 7)
getw (9, root) → 11
resetw ((6,7), ...)
resetw ((6,10), ...)
addw ((10,14), ..., 3)
getw (9, root)
u.e
u.w
u.ls
u.rs
ls(u)
rs(u)
[1,17)
[1,9)
[9,17)
[1,5)
[5,9)
[9,13)
[13,17)
[1,3)
[3,5)
[5,7)
[7,9)
[9,11)
[11,13)
[13,15)
[15,17)
[1,2)
[2,3)
[3,4)
[4,5)
[5,6)
[6,7)
[7,8)
[8,9)
[9,10)
[10,11)
[11,12)
[12,13)
[13,14)
[14,15)
[15,16)
[16,17)
4
7
4
4
4

Basic operations performed on the weighted segment trees include

addw: 'adding a weight to a given interval'

resetw: 'resetting (to 0) the weight of a given interval' and

getw: 'getting the accumulated weight of a given elementary interval.'

```
resetw(v:interval,u:node,wsum:int)
   begin
     if u is reset-marked then begin
             ls(u).w =rs(u).w =0
             mark ls(u) and rs(u)
             unmark u
     end
     if (v.B <= u.B and u.E <= v.E)
     then begin
             mark u
             u.w = -(wsum);
          end
     else begin
             wsum = wsum + u.w
             if (v.B < u.M) then
             resetw(v,ls(u),wsum)
             if (u.M < v.E)
             resetw(v,rs(u),wsum)
          end
   end
```

```
addw(v:interval,u:node,weight:int)
   begin
     if u is reset-marked then
     begin
        ls(u).w =rs(u).w =0
        mark ls(u) and rs(u)
        unmark u
     end
     if (v.B <=u.B and u.E <=v.E)
     then u.w = u.w + weight
     else begin
          if (v.B < u.M) then
          addw(v,ls(u),weight)
          if (u.M < v.E) then
          addw(v,rs(u),weight)
         end
   end

getw(ev:int,u:node)
   begin
     if (u.B =ev and u.E =ev + 1)
     then return(u.w)
     if (u is reset-marked)
     then return(u.w)
     if (ev<u.M) then
        return(u.w+getw(ev,ls(u)))
     if (ev>u.M) then
        return(u.w+getw(ev,rs(u)))
   end
```


20—

Path Weight Restructuring in Communication Graphs

Michael Lightner
University of Colorado
Boulder, Colorado

The abstract and transparencies for this talk were not available.

Assessing Similarity of Pathfinder Graphs

Daniel M. Davenport
Timothy E. Goldsmith
Peder J. Johnson

December 1, 1989

1 Abstract

In this talk we present several graph similarity measures for assessing the similarity of Pathfinder graphs. This work was motivated by the desire to measure the similarity between a graph representing a student's knowledge (*a knowledge structure*) and that of the instructor's. Our measures compare connected, labelled graphs, such as knowledge structures and other graphs produced by Pathfinder. Other graph similarity measures have been proposed for various other applications but none are suitable for our purposes.

We first give some background and make the hypothesis that knowledge structures model a student's knowledge of a subject. To verify this we define several graph similarity measures and with them measure the similarity of a student's knowledge structure with that of the instructor's. We then correlate this with that of the student's final grade. The resulting positive correlations verify our hypothesis. We also show that neighborhoods in knowledge structures are a more important feature for modeling knowledge than distance between nodes. The success of our measures in predicting students' final grades gives us hope that these measures have applicability to proximity graphs in general.

Assessing Similarity of Pathfinder Graphs

Daniel M. Davenport
Timothy E. Goldsmith
and
Peder J. Johnson

Talk Outline

1. Introduction and Background

2. Approaches to Graph Similarity Measures

3. Definitions of our Graph Similarity Measures

4. Application to Knowledge Structures

5. Properties of our Graph Similarity Measures

6. Topics for Further Research.

Background

* Students in a class are asked to rate the similarity of pairs of concepts they've learned.

* For each student these raw similarity ratings are fed into the Pathfinder algorithm.

* The result is a connected, unweighted graph, known as a knowledge structure, for each student.

* An instructor can visually compare his own knowledge structure to that of a student's and can sometimes pick out the students that know the subject.

Hypothesis

* The pattern of edges of a knowledge structure models what a student knows.

* That is, Pathfinder extracts from the raw similarity ratings important features of a student's knowledge.

The Problem

* Find an objective way to compare an instructor's *knowledge structure* to that of a student's.

*That is, find a function that takes two graphs and returns a number that reflects their "closeness".

* Such a function is a *graph similarity* measure.

* A *good* graph similarity measure is one that verifies our hypotheses.

* Thus, we must find a good graph similarity measure.

* Keep it simple.

Some Graph Similarity Measures in the Literature

* Herndon
  * Compares molecules.
  * Computes longest paths in graphs (hard to do).
  * Paths are converted to linear codes which are then compared.

* Basak et al.
  * Compares molecules.
  * Measures similarity of graph-theoretic indices of each graph (such as, the number of nodes, the degree of sequence, the number of paths of length k).
  * Combines this data using complicated information-theoretic techniques.

* Graham, Ulam
  * Compares abstract graphs.
  * Graphs must have same number of edges.
  * Partitions graphs into minimal number of pairwise isomorphic pieces.

Our Approach

* The graphs we wish to compare are:
  * connected
  * unweighted
  * and have a common node set.

* Thus, we already know which nodes correspond between the graphs.

* This suggests three approaches:

  * Base the measure on the similarity of the (nonempty) set of neighbors of corresponding nodes.

  * Base the measure on the similarity of the incidence of pairs of corresponding nodes.

  * Base the measure on the similarity of the minimal path length between pairs of corresponding nodes.

Preliminary Definitions

Let G be a connected graph and v, v' nodes in G.

Define G_v to be the set of nodes in G that are neighbors of v (i.e., incident with v).

Note that G_v is not empty since G is connected and note also that v is not in G_v.

Define G(v,v') to be 1 if v is incident with v' and 0 otherwise.

Define $\delta_G(v,v')$ to be the distance from v to v' in G.

This is always defined since G is connected and is never 0.

For x, y > 0, define $x \,\theta\, y$ to be x/y if $x \leq y$ and y/x otherwise.

Let A and B be connected graphs with a common node set V. Suppose further that the elements of V are linearly ordered.

Neighborhood Based Measures

$$C_1(A,B) = \frac{1}{|V|}\sum_{v\in V}\frac{|A_v \cap B_v|}{|A_v \cup B_v|}$$

$$C_2(A,B) = \frac{1}{|V|}\sum_{v\in V}\frac{|A_v \cap B_v|}{(|A_v| + |B_v|)/2}$$

$$C_3(A,B) = \frac{1}{2}\left(\frac{1}{|V|}\sum_{v\in V}\frac{|A_v \cap B_v|}{|A_v|} + \frac{1}{|V|}\sum_{v\in V}\frac{|A_v \cap B_v|}{|B_v|}\right)$$

Incidence Based Measures

$$C_4(A,B) = 1 - \frac{1}{(|V|^2 - |V|)/2} \sum_{v<v'} | A(v,v') - B(v,v') |$$

$$\begin{aligned} C_7(A,B) = {} & \text{Correlation coefficient of } A(v,v') \text{ and } B(v,v') \\ & \text{for all pairs of nodes } v, v' \text{ with } v < v' \end{aligned}$$

Distance Based Measures

$$C_5(A, B) = \frac{1}{(|\ V\ |^2 - |\ V\ |)/2} \sum_{v<v'} \delta_A(v, v')\ \theta\ \delta_B(v, v')$$

$$C_6(A, B) = 1 - \frac{1}{(|\ V\ |^2 - |\ V\ |)/2} \sum_{v<v'} \frac{|\ \delta_A(v, v') - \delta_B(v, v')\ |}{\delta_A(v, v') + \delta_B(v, v')}$$

$$\begin{aligned} C_8(A, B) = {} & \text{Correlation coefficient of } \delta_A(v, v') \text{ and } \delta_B(v, v') \\ & \text{for all pairs of nodes } v, v' \text{ with } v < v' \end{aligned}$$

Testing our Hypothesis

* 20 students and the instructor of a class assessed the similarity of 30 concepts from the class.

* 435 pairs of concepts were rated on a scale from 1 (least similar) to 7 (most similar).

* These raw similarity ratings were processed by Pathfinder to produce a knowledge structure for each individual.

* For each C_i, the similarity of each student's knowledge structure and the instructor's were measured and then correlated with the student's final course grade.

The Correlations

Neighborhood Based Measure	$C_1 = .77$
Incidence Based Measure	$C_4 = .38$
Distance Based Measure	$C_8 = .65$

Conclusion: The patterns of edges of a knowledge structure model what a student knows if final course grades do.

New Hypothesis

* Neighborhood based measures are better for assessing a student's knowledge than distance based or incidence based measures.

* That is, neighborhoods of concepts in knowledge structures are a more important feature than distance or incidence in modeling knowledge with knowledge structures.

Testing our Hypothesis

* Using partial correlations we can remove the shared contribution of a measure from every other measure and thereby examine the unique predictiveness of the first measure.

Partial Correlations

Removing C_4 from C_1 - .73

Removing C_8 from C_1 - .53

Removing C_1 from C_4 - .09

Removing C_8 from C_4 - .11

Removing C_1 from C_8 - .14

Removing C_4 from C_8 - .57

Conclusion: Neighborhoods of concepts in knowledge structures are a more important feature than distance or incidence in modeling knowledge with knowledge structures.

Further Properties

* For each C_i let $D_i=1-C_i$. Then D_1, D_2, D_4, D_5, and D_6 are all metrics on the space of graphs with a common node set V, while D_3 is not.

* We can think of graphs with a common node set V as subsets of VxV. Form the Boolean ring obtained by defining multiplication by intersection and addition by symmetric difference. The multiplicative identity, I, is the completely connected graph and the zero element is the graph with no edges.

If we define $\frac{0}{0} = 1$ then D_1 is a metric on this space.

For a graph G over V define:

$$\| G \| = 1 - D_1(G, I) = C_1(G, I)$$

Then for all A, B, C graphs over V we have:

$$\| A \cup B \| + \| A \cap B \| = \| A \| + \| B \|$$

$$1 - \| \bar{A} \| = \| A \|$$

$$\| A \oplus B \| \leq \| A \| + \| B \|$$

$$\| A \cap B \| \leq \| A \|$$

$$\| A \oplus B \| + \| A \oplus C \| \leq \| B \oplus C \|$$

where $\bar{A}$ is the complement of A.

It turns out that $\| A \oplus B \| = D_4(A, B)$.

Topics for Further Research

* Generalize our measures to measure similarity of molecules.

* Find applications of these measures to general proximity graphs.

* Find features of knowledge structures other than neighborhoods that are important in assessing student's knowledge.

* Develop a neighborhood based graph clustering technique.

ABSTRACT

Generating Large Pathfinder Networks

Govinda Kurup
Xerox Corporation
Webster, NY 14580

A geometrical approach to the Pathfinder algorithm which reduces the actual number of computations by half is presented. In the original algorithm, for every pair of nodes the path length via an intermediate node is computed. The number of intermediate nodes considered is n-2 (where n is the total number of nodes), for each pair. The essence of the current approach lies in presorting the distance matrix by which the number of intermediate nodes inspected is reduced to (n-2)/2. The method is general and works for any value of the r-metric and q-parameter. The saving in actual computation time for large PFNs is substantial.

The transparencies for this talk were not available.

ABSTRACT

Integrity Considerations in Graphs

Lowell W. Beineke

Indiana University - Purdue University at Fort Wayne

The vertex-integrity of a graph is defined to be $I(G) := \min (|X| + m(G-X))$, where the minimum is taken over all proper subsets X of the vertex set and m(G-X) denotes the largest order of a component of G-X. The edge-integrity I'(G) is defined similarly, and both parameters are measures of a graph's vulnerability to disruption when elements of the graph are destroyed. This talk presents aspects of integrity that might be useful in analyzing proximity graphs, and in particular these topics: (1) bounds and algorithms for trees and other planar graphs, and (2) the diameter of a graph.

A TALK OF INTEGRITY

EDGE-INTEGRITY AND DIAMETER

L. W. BEINEKE*

W. D. GODDARD

M. J. LIPMAN

K. S. BAGGA

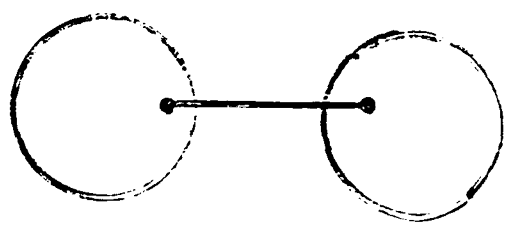
R. E. PIPPERT

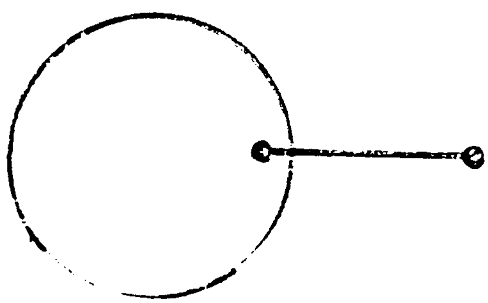
HOW SHOULD THE VULNERABILITY OF A COMMUNICATION NETWORK BE MEASURED?

ONE WAY IS TO USE THE CONNECTIVITY OR THE EDGE-CONNECTIVITY.

THESE ARE NOT SENSITIVE TO WHAT REMAINS AFTER KEY ELEMENTS ARE DESTROYED.

FOR EXAMPLE, $\lambda = 1$:





IF ONE IS INTERESTED IN DISRUPTING COMMUNICATION, IT IS DESIRABLE TO HAVE TWO QUANTITIES SMALL:

(i) THE NUMBER OF ELEMENTS DESTROYED

(ii) THE SIZE OF THE LARGEST GROUP THAT CAN STILL COMMUNICATE.

BAREFOOT, ENTRINGER, AND SWART INTRODUCED A MEASURE OF HOW SUCCESSFULLY THIS CAN BE DONE.

(VERTEX-) INTEGRITY

$$I(G) := \min_{X \subset V} \{ |X| + m(G-X) \}$$

WHERE $m(G-X)$ IS THE ORDER OF A LARGEST COMPONENT OF $G-X$.

$I(K_p) = p$ $\qquad$ $I(\overline{K}_p) = 1$

$I(K_{1,n}) = 2$ $\qquad$ $I(K_{r,s}) = 1 + r$ if $r \leq s$.

Consider P_p, the path with p vertices.

If r vertices are removed, then $r+1$ or fewer components remain, and so one has at least $\frac{p-r}{r+1}$ vertices. Hence

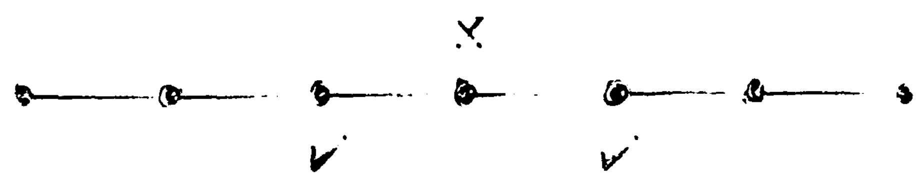
$$I(P_p) \geq \min_r \left\{ r + \frac{p-r}{r+1} \right\}$$

For $x \geq 0$, $f(x) := x + \frac{p-x}{x+1}$ has min value of $2\sqrt{p+1} - 2$. Therefore

$$I(P_p) \geq \lceil 2\sqrt{p+1} \rceil - 2.$$

This can be achieved in all cases.

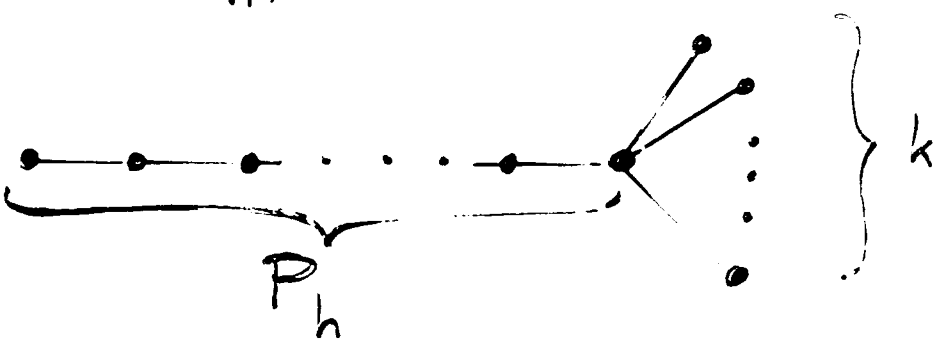
P_7 :



Integrity of Trees

For any T_p $\quad I(K_{1,p-1}) \le I(T_p) \le I(P_p$

Comet $C_{h,k}$



Moving a vertex from the head to the tail increases I by at most 1, so comets achieve all intermediate values.

$$I(C_{h,k}) = \begin{cases} \lceil 2\sqrt{p+1}\,\rceil - 2 & \text{if } k \le \sqrt{p+1} - \\ \lceil 2\sqrt{h}\,\rceil - 1 & \text{otherwise} \end{cases}$$

$(h+k=p)$
$(h \ge 2)$

The range of $I(T_p)$ is $\{2, 3, \cdots, I(P_p)\}$.

Graph G	$I(G)$
K_p	p
$\overline{K_p}$	1
P_p	$\lceil 2\sqrt{p+1}\rceil - 2$
C_p	$\lceil 2\sqrt{p}\rceil - 1$
$K_{m,n}$	$1 + \min\{m,n\}$
- - -	
Q_n	? $1+2^{n-1}$

G & S

(a) $I(G) = 1$ iff G is null.

(b) $I(G) = 2$ iff for some $r \geq 0$, $s \geq 1$
$G \cong rK_1 \cup sK_2$ or $rK_1 \cup K_{1,s}$.

(c) $I(G) = p$ iff G is complete.

(d) $I(G) \geq p-1$ iff $\overline{G}$ has girth $\gamma \geq 5$.

I-*minimal* : $I(G-e) < I(G)$ $(\forall e \text{ in } G)$

Complete graphs are I-minimal.
Every graph has I-minimal subgraph with same I.
K_2 is the only one of integrity 2.

I-*critical* : $I(G-v) < I(G)$ $(\forall v \text{ in } G)$

I-minimal $\Rightarrow$ I-critical
(no isolates)

I-critical but not I-minimal : C_{n^2}

I-*maximal* : $I(G \cup e) > I(G)$ $(\forall e \text{ in } \bar{G})$

$\Longleftrightarrow$ $K_r + (K_{n_1} \cup \cdots \cup K_{n_t})$, $t \geq 2$ and

with $n_1 \leq \cdots \leq n_t$, $n_{t-1} = n_t \leq n_1 + n_2 - 1$.

I-*acritical* : $I(G-v) = I(G)$ $(\forall v \text{ in } G)$

Integrity and other parameters

(a) $I(G) \le \alpha(G) + 1$ covering nr.

$=$ iff $2K_2$ not induced

(b) $I(G) \ge \delta(G) + 1$ min. deg.

$=$ iff rK_n or $rK_n + F$, [illegible] with $\delta(F) \ge$ [illegible]

(c) $I(G) \ge \min\limits_{t} \max\{d_t, t-1\}$ if $d_1 \ge \cdots \ge d_p$ degrees

(d) $I(G) \ge \chi(G)$ chromatic nr.

(e) $I(G) \ge \dfrac{p - \kappa(G)}{\beta(G)} + \kappa(G)$ conn'ty indep. nr.

(f) $I(G) \ge 2\sqrt{\tau p} - \tau$ toughness
(not K_p)

Nordhaus-Gaddum

(a) $I + \overline{I} \geq p+1$

(b) $I \cdot \overline{I} \geq p$

(b) is sharp only for complete or null

(a) is sharp for others too ($K_{m,n}$, $K_{2,4}-e$)

Def: $r(m,n)$ = Ramsey no.: min no. of [illegible] so all graphs have m mutually adj. [illegible] or n independent.

Let $S_p := 2p + 4 - \min\{m+n : r(m,n) > p\}$

(a) $I + \overline{I} \leq S_p$

(b) $I \cdot \overline{I} \leq S_p^2/4$

(a) sharp for $p \leq 10$ except $p=8$

The biggest known $I + \overline{I}$ (for $p > 4$) is $\left\lceil \frac{3p}{2} \right\rceil$ -- for $C_p^{\lfloor p/4 \rfloor}$.

Binary Operations

1. If cpts are G_i, then

$$\max I(G_i) \le I(G) \le \sum I(G_i) - n + 1.$$

2. $$I(G+H) = \min\{ I(G) + H,\ I(H) + G \}$$

3. $I(G[H])$ is known; in particular

$$I(G[K_n]) = n\, I(G);$$

$$I(K_n[G]) = (n-1)|G| + I(G).$$

4. $$I(|G|\, H) \le I(G \times H) \le I(G[H])$$

$$\tfrac{3}{2} I(G) \le I(K_2 \times G) \le 2 I(G)$$

For $n \ge 5$, if $n = r^2 + k$ with $0 \le k \le 2r$

$$I(K_2 \times C_n) = \begin{cases} 2\, I(C_n) - 1 & \text{if } 1 \le k \le \frac{r}{2} \\ & \text{or } r < k \le \frac{3r}{2} \\ 2\, I(C_n) & \text{otherwise} \end{cases}$$

$I(K_2 \times P_n)$ is similar

$$I(Q_n) = ?$$

Mean Integrity (Chartrand, Kapoor, McKee, Oellermann)

$p_v(G)$:= order of component containing v

$\overline{m}(G) := \frac{1}{p} \sum_v p_v(G)$ $\quad$ [$m(G) := \max p_v(G)$

$$J(G) := \min_{S \subseteq V} \{ |S| + \overline{m}(G-S) \}$$

$J(G) \le I(G)$, = for many elementary graphs

For every rational $r \ge 1$, there is G with $J(G) = r$. Q: Range for fixed order?

Some results on bounds are known.

Goddard Schema

For a given parameter ψ, define Ψ

$$\Psi(G) := \min_{X \subset V} \{|X| + \psi(G-X)\}$$

Examples: I, J, K $\left(\psi = \begin{array}{l} |G|-1 \text{ if conn.} \\ 0 \text{ otherwise} \end{array}\right)$

Let $\psi_0 = \psi(K_1)$.

$$\Psi + \overline{\Psi} \geq p - 1 + 2\psi_0$$

$$\Psi \cdot \overline{\Psi} \geq p\,\psi_0$$

The range of Ψ on graphs of order p:

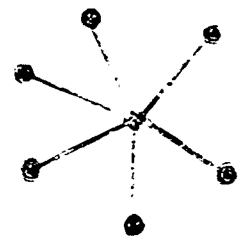
$$\Psi(\overline{K}_p), \quad \ldots, \Psi(K_p)$$

EDGE-INTEGRITY

$$I'(G) := \min_{S \subseteq E} |S| + m(G-S)$$

where $m(G) :=$ order of a largest compon

Stars $\quad I'(K_{1,s}) = s+1$



Paths $\quad I'(P_n) = \lceil 2\sqrt{n} \rceil - 1$

If H is a subgraph of G, then $I'(H) \le I$

Theorem. For any tree T_p, $I'(T_p) \ge \lceil 2\sqrt{p}$

If r satisfies $\lceil 2\sqrt{p} \rceil - 1 \le r \le p$, then there exists tree of order p with $I'(T) = r$.

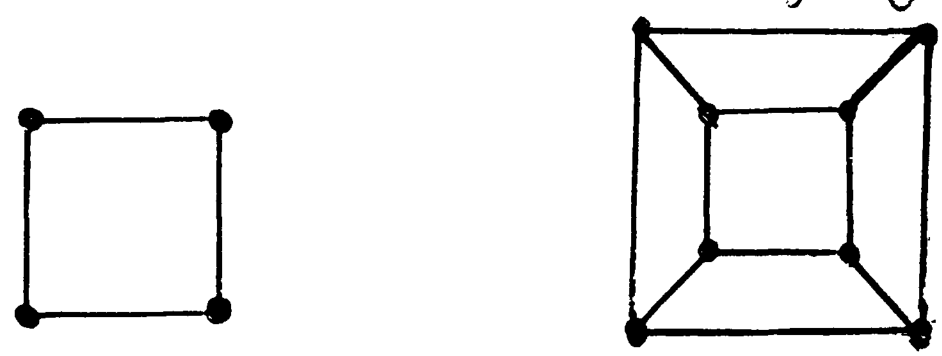
Note that, among trees, stars have the least integrity and paths the greatest, but as regards edge-integrity, the reverse is true.

In general, $I(G) \le I'(G)$;
for trees, if equality, then $p = n^2$ or $n(n+1)$.

$$I'(G) \ge \Delta + 1 \qquad I(G) \ge \delta + 1$$

Call G <u>honest</u> if $I'(G) = p$.

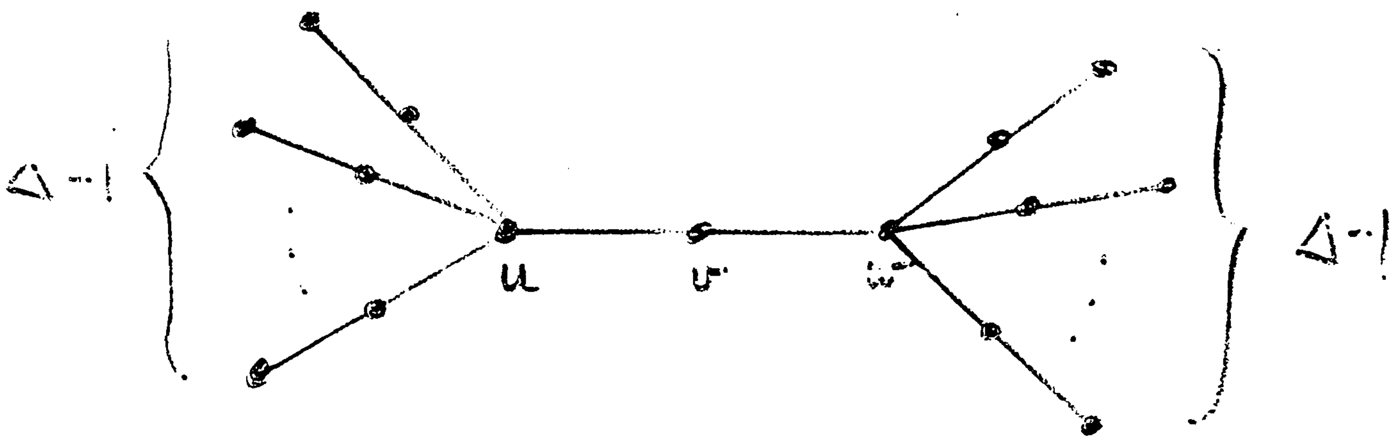
Examples: Stars, any graph with a vertex of deg $p-1$.



THEOREM. (a) If $\Delta \geq \frac{1}{2}p$, then $I'(T) = \Delta + 1$.

(b) If $\Delta \leq \frac{1}{2}p$, then $I'(T) \leq \frac{1}{2}(p+3)$.

For $\frac{1}{4}(p+1) \leq \Delta \leq \frac{1}{2}p$, (b) is best possible.

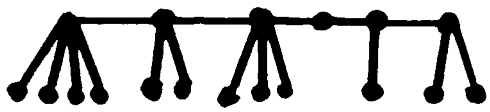


$p = 4\Delta - 1$

$I' = 2\Delta + 1$ $\qquad |S| = 2(\Delta - 1)$

$m(G-S) = 3$

Algorithms.

Caterpillars 

$$\Delta + 1 \leq I' \leq p$$

For m in this range, $\varphi(m)$ can be computed by taking those spinal edges in succession which would result in too large a component if not taken — take no end edges.

Associate with a spine vertex v number of end-edges at v; work with this sequence $(a_1, \cdots, a_r)$ (or your own variation).

I'(T) Algorithm

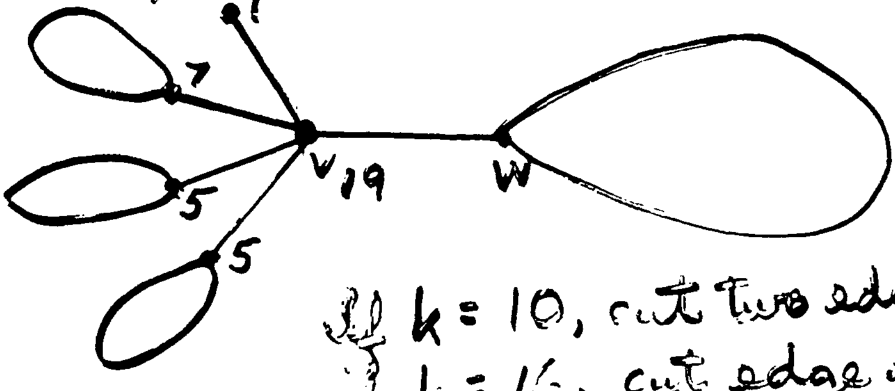
Note: By taking S minimal, never use an end-edge.

For each possible m achieving I' = (m + s), find an optimal I'-set S(m):

1. Weight each end vertex 1. Consider k.
2. Take an unweighted vertex v having at most one unweighted neighbor. Make its weight w: sum of nbr-wts + 1.

   If w ≤ k, continue.

   If w > k, cut as few edges as possible at v to get sum at most k, and small.

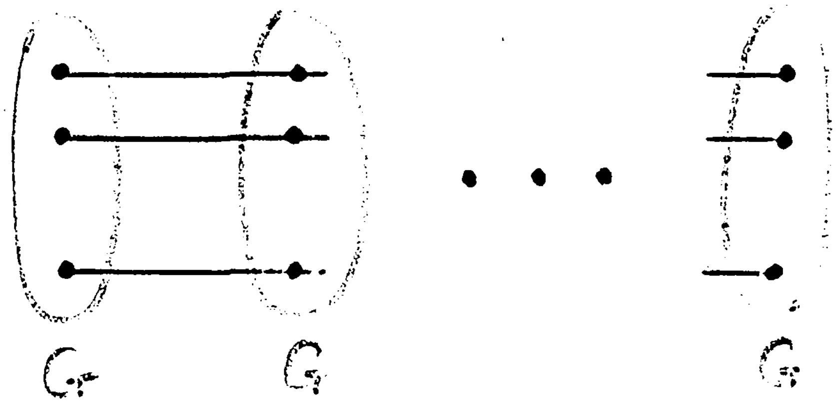


If k = 10, cut two edges;
If k = 16, cut edge at 7;
If k = 23, continue.

Cartesian products

$G \times H$ Take $|G|$ copies of H and join corresponding vertices of H for the edges of G.

$P_n \times G$



THM. $I'(K_n \times G) = n\, I'(G).$

COR. $I'(Q_n) = 2^n$

(Cubes are honest.)

THM.

$$(\delta(G)+1) \cdot I'(H) \le I'(G \times H) \le |V(G)| \cdot I'(H)$$

A method for lower bounds.

DEFS. For $X \subset V$, $\deg X :=$ number of edges joining X to $V-X$.

For $k \in \mathbb{N}$, $\beta_G(k) := \min_{|X|=k} \deg X$

Let G be a finite graph of order p which is an induced subgraph of H, which is possibly infinite.

Let $\varphi_H(G) := \min\{M + \frac{1}{2}(\sum_1^k \beta(m_i) - \deg V(G))\}$,

where the minimum is taken over all $M \geq 0$

and all partitions $m_1 + \cdots + m_k = p$, $0 < m_i \leq M$.

THM. $I'(G) \geq \varphi_H(G)$.

Grid Graphs

$$G = P_r \times P_s \qquad H = P_\infty \times P_\infty$$

$$\deg(P_h \times P_k) = 2(h+k)$$

$$\beta_H(p) \geq 4\sqrt{p}$$

$$I'(P_r \times P_s) \geq \min_{x>0} \; x + \tfrac{1}{2}\sum_{1}^{\ell} \beta_H(n_i) - (r+s)$$

where $n_1 + n_2 + \cdots + n_\ell = rs, \quad 0 < n_i \leq x.$

Middle term:

$$\tfrac{1}{2}\sum \beta_H(n_i) \geq 2\sum\sqrt{n_i} \geq 2\frac{rs}{\sqrt{x}}$$

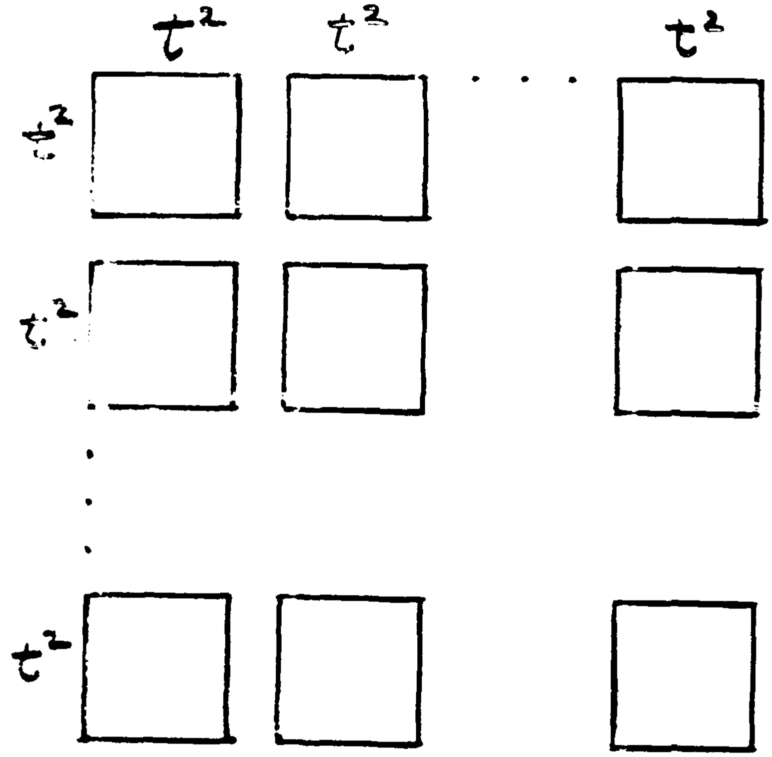
If $f(x) := x + \dfrac{2rs}{\sqrt{x}} - (r+s)$,

then $f'(x) = 1 - \dfrac{rs}{x^{3/2}}$, so min at $x = (rs)^{2/3}$.

$$I'(P_r \times P_s) \geq (rs)^{2/3} + \frac{2rs}{(rs)^{1/3}} - (r+s)$$

THM. $I'(P_r \times P_s) \geq 3(rs)^{2/3} - (r+s)$.

Ex. 1. $r = s = t^2$ Lower Bound $3t^4 - 2t^3$



$m = t^4$

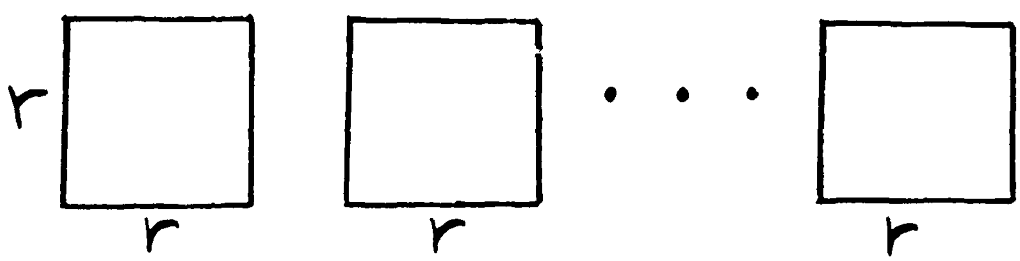
$|S| = 2(t-1)t^3$

Upper Bound

$t^4 + 2(t^4 - t^3)$

Ex. 2. $s = r^2$ $I'(P_r \times P_{r^2}) = 2r^2 - r$

$= r\, I'(P_r)$



r *blocks*

Question. $I'(P_n \times G) \overset{?}{\sim} |G| \cdot I'(P_n)$

Paths, Cycles, Stars (one small)

$I'(P_3 \times P_n) = 3\, I'(P_n)$

$I'(P_5 \times P_5) < 5\, I'(P_5)$

$I'(P_4 \times P_n)$?

$I'(C_3 \times G) = 3\, I'(G)$ $(C_3 = K_3)$

$I'(C_4 \times G) = 4\, I'(G)$ $(C_4 = K_2 \times K_2)$

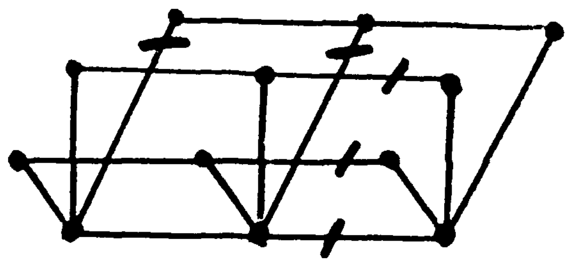
$I'(C_5 \times C_n) = 5\, I'(C_n)$

$I'(P_3) = 3$

$I'(K_{1,3}) = 4$

$I'(P_3 \times K_{1,3})$

$= 11$



Complements

diam $G = 2 \Rightarrow G$ honest $(I'(G) = |G|)$

G or $\overline{G}$ has diam ≤ 3

THM. Unless G is P_4, G or $\overline{G}$ is honest.

(IPFW; [illegible], Stickles, Riggs)

Nordhaus-Gaddum

THM. For $p \geq 5$,

(a) $p + 1 \leq I'(G) + I'(\overline{G}) \leq 2p$

(b) $p \leq I(G) \cdot I'(\overline{G}) \leq p^2$

Upper bounds sharp since there exist honest graphs with honest complements.

EDGE-INTEGRITY AND DIAMETER

$$I'(G) = \min_{S \subseteq E(G)} \{ |S| + m(G-S) \}$$

THEOREM. If G has diameter 2, it is honest.

Proof: Let S be an I'-set of minimum cardinality. If $S = \phi$, then $m(G-S) = p$, so we assume $S \neq \phi$.

By the minimality of S, every edge in S joins different components of $G-S$.

Three cases:

(i) Some vertex is on no S-edge.

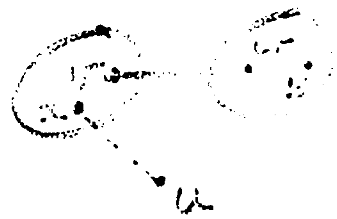
(ii) Every vertex is on at least one S-edge and some vertex is on only one.

(iii) Every vertex is on more than one S-edge.

Case (i). Assume v is on no S-edge. Then v and all its neighbors are in one component C of $G-S$. If w is not in C, then there is a $v-w$ 2-path vuw and edge uw must be in S. Hence $|S| \geq p - |V(C)|$, Since $m(G-S) \geq |V(C)|$,

$$I'(G) = |S| + m(G-S) \geq p.$$

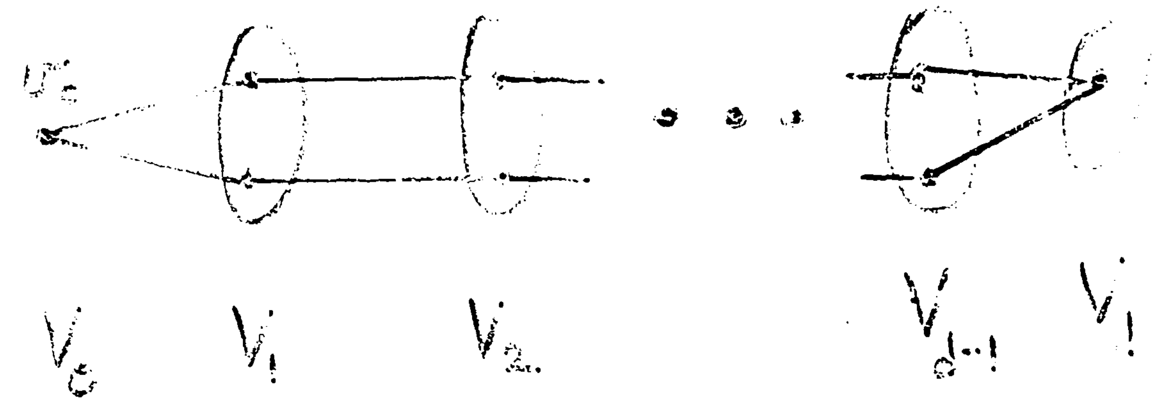
Case (ii). Assume every vertex is on an S-edge and v is on exactly one, say $e = vw$. Consider components C_v and C_w. If u is in neither, there must be a 2-path vxu and xu must be in S. If x is in C_w, then $x = w$. It follows that each y in C_w is on an S-edge different from these and each other. Hence, again $|S| \geq p - |V(C_v)|$, as before.



Case (iii). Every vertex is on more than one S-edge. Then $|S| \geq p$. //

LET G BE HONEST, WITH DIAMETER d.

LET v_0 BE A 'PERIPHERAL' VERTEX.



LET $a_i = |V_i|$, $A_n = \sum_0^n a_i$

LET r BE LARGEST SO $A_r \le P/2$.

SINCE G IS HONEST, IT MUST NOT 'PAY' TO CUT THE EDGES BETWEEN V_n, V_{n+1}.

HENCE, FOR $n = 0, 1, \cdots, r-1$,

$$A_n \le a_n a_{n+1};$$

AND FOR $n = r, \cdots, d-1$,

$$p - A_n \le a_n a_{n+1}$$

<u>Lemma.</u> Let $\{b_i\}$ be a sequence with partial sums B_n such that $b_i \geq 1$ and $b_i\, b_{i+1} \geq B_i$ for all i. Then $B_{2k-1} \geq k(k+1)$ and $B_{2k} \geq (k+1)^2$.

Proof: Induction on k. — Even case.
Assume $B_{2k} \geq (k+1)^2$. Then

$$b_{2k+2} \geq B_{2k+1} / b_{2k+1} = \frac{B_{2k}}{b_{2k+1}} + 1.$$

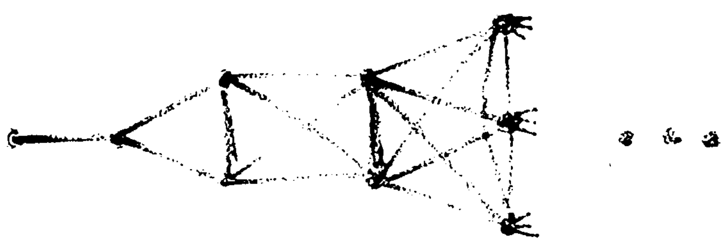
Let $b_{2k+1} = x$. Then

$$\begin{aligned} B_{2k+2} &= B_{2k} + b_{2k+1} + b_{2k+2} \\ &\geq B_{2k} + x + \frac{B_{2k}}{x} + 1 \\ &\geq (k+1)^2 + x + \frac{(k+1)^2}{x} + 1 \\ &\geq (k+1)^2 + (k+1) + (k+1) + 1 = (k+2)^2 \end{aligned}$$

(since $x = k+1$ gives min). //

The minimum is achieved for

$$a_0 = a_1 = 1$$
$$a_2 = a_3 = 2$$
$$a_4 = a_5 = 3$$
$$\vdots$$



Assume $d = 4t \quad (m = 2t)$

$$\alpha_{2t} \geq (t+1)^2$$
$$p \geq 2(t+1)^2$$
$$p \approx d^2/8 \,.$$

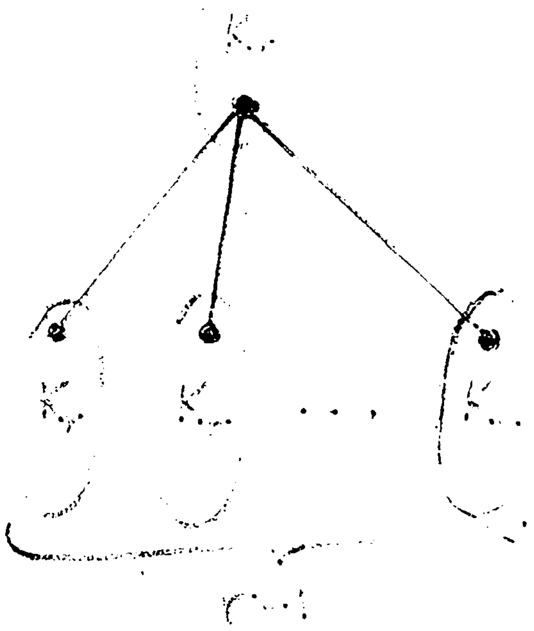
THEOREM. THE MINIMUM ORDER OF AN HONEST GRAPH OF DIAMETER d IS

$$p = \begin{cases} 2n^2 & \text{IF } d = 4n-3 \\ 2n^2 + n + 1 & \text{IF } d = 4n-2 \\ 2n^2 + 2n + 1 & \text{IF } d = 4n-1 \\ 2n^2 + 3n + 1 & \text{IF } d = 4n \end{cases}$$

THE GREATEST DIAMETER OF AN HONEST GRAPH WITH p VERTICES IS ABOUT $\sqrt{8p}$.

DIAMETER 4.

CONSIDER G_{r^2}:



$I'(G_{r^2}) \leq 2r-1$

$I'(P_{r^2}) = \lceil 2\sqrt{r^2} \rceil - 1 = 2r-1.$

FOR $p \geq 5$, THERE IS A GRAPH OF ORDER p AND DIAMETER 4 WHOSE EDGE-INTEGRITY EQUALS THAT OF THE PATH OF ORDER p.

Coloring Proximity Graphs

Bob Cimikowski
New Mexico State University

Abstract

We examine coloring problems for various proximity graphs. The *chromatic number problem* is that of finding the minimum number of colors to assign to the vertices of a graph so that adjacent vertices have different colors. The *minimum coloring problem* is to find a minimum assignment of colors for a graph. Both are hard problems for arbitrary graphs as well as planar graphs. For proximity graphs, the problems have applications in transmitter frequency assignment and event scheduling. The problems are also of theoretical interest in the field of algorithmic computational complexity.

The proximity graphs investigated are Relatively Closest graphs, Relative Neighborhood graphs, Gabriel graphs, and Delaunay graphs. We restrict the graphs to 2-dimensional Euclidean space, for which they are all planar.

Our results include a linear-time test for the chromatic number of a Delaunay graph and a linear 3-coloring algorithm, an exact linear 4-coloring algorithm for Relatively Closest graphs, a 4-coloring heuristic for Relative Neighborhood graphs with remarkably good performance, and two minimum-coloring heuristics for Gabriel graphs which outperform other methods on the same set of test graphs.

We conclude with a number of open problems and suggestions for further research.

Coloring Proximity Graphs

1. Theoretical Issues.
2. Applications.
3. Delaunay graphs.
4. Relative Neighborhood graphs.
5. Relatively Closest graphs.

Applications of Proximity Graph Coloring:

1. Minimum frequency assignment

   - n data sampling stations, transmitting at same power.
   - neighbors must have different frequencies to avoid interference.

2. Event scheduling at geographic sites.

   - events cannot occur simultaneously at neighboring sites.

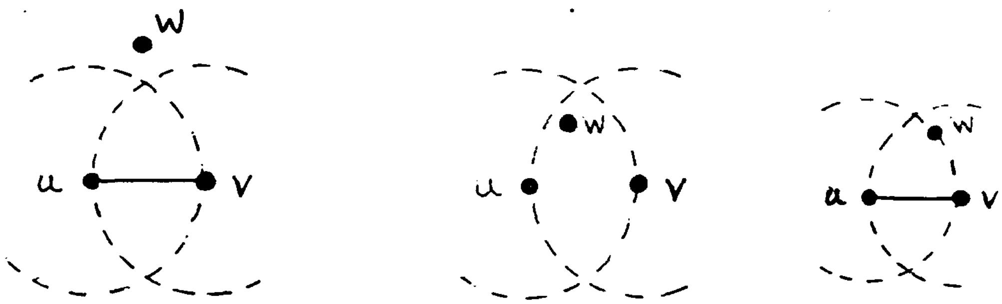
Families of Proximity Graphs

Euclidean distance metric:

$$d(u,v) = \sqrt{(u_1 - v_1)^2 + (u_2 - v_2)^2}$$

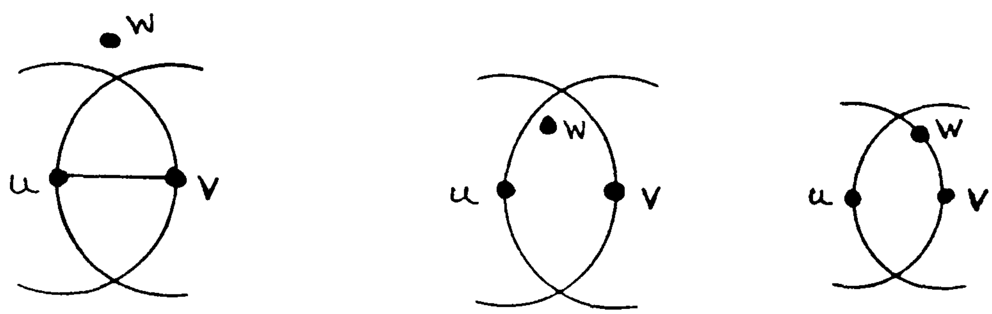
Relative Neighborhood Graph (RNG)

"lune of influence (open)"



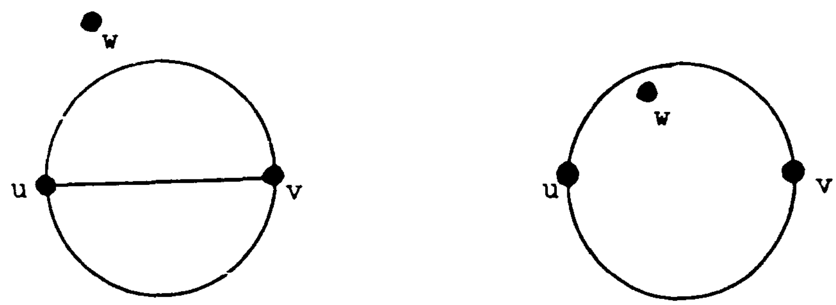
Relatively Closest Graph (RCG)

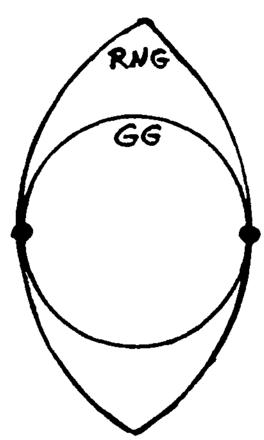
"lune of influence (closed)"



Gabriel Graph (GG)

"circle of influence"





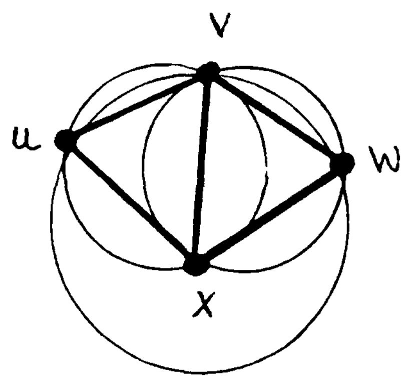
Delaunay Graph (DG)

Voronoi Diagram - partition of plane and points P into polygons $V(p)$, for each $p \in P$.

$$V(p) = \{x : d(x,p) < d(x,q), \text{ for } each\ p \neq q,\ q \in P\}.$$

Delaunay Graph = straight-line dual of Voronoi Diagram.

$\{u,v\}$,$\{v,w\}$, and $\{u,w\}$ are edges of the DG iff $circle(u,v,w)$ is empty.



Combinatorial Properties of Proximity Graphs

P1. Planar; $e \leq 3n - 6$.

P2. DG is inner-triangulated and 2-connected.

P3. RCG forbidden subgraphs: K_3, $K_{2,3}$.

P4. RNG forbidden subgraphs: K_4, $K_{2,3}$, W_n, $n \leq 5$.

P5. GG forbidden subgraphs: K_4, $K_{2,3}$, W_n, $n \leq 5$.

Theorem 1 [Toussaint]. For any nondegenerate set of points V,

1) $MST(V) \subseteq RNG(V) \subseteq GG(V) \subseteq DG(V)$.

2) $RCG(V) \subseteq RNG(V) \subseteq GG(V) \subseteq DG(V)$.

Theorem 2 [Haken and Appel]. Every planar graph is 4-colorable.

Theorem 3 [L. Stockmeyer]. 3-colorability is NP-complete for planar graphs.

Fact. Any planar graph can be 4-colored in $\Theta(n^2)$ time.

-- but the method is impractical!

Fact. Any planar graph can be 5-colored in linear time.

Theorem 4.1. [Saint-Lague]. A maximal planar graph is 3-colorable iff all vertices have even degree.

Theorem 4.2. [Cimikowski]. A Delaunay graph G is 3-colorable iff all interior vertices have even degree.

--implies a linear-time test for $\chi(G)$.

Fact. Any 3-colorable Delaunay graph G is *uniquely* 3-colorable (i.e., every 3-coloring induces same partition of V(G)).

--leads to linear-time 3-coloring algorithm.

Open Problems:

1. Complexity of Gabriel graph 3-colorability?

2. Complexity of Relative Neighborhood graph 3-colorability?

   Conjecture: both NP-complete.

   Proof: difficult without a combinatorial characterization of the graphs.

4-Coloring Heuristic for Relative Neighborhood graphs

Fact. The minimum degree of any planar graph ≤ 5.

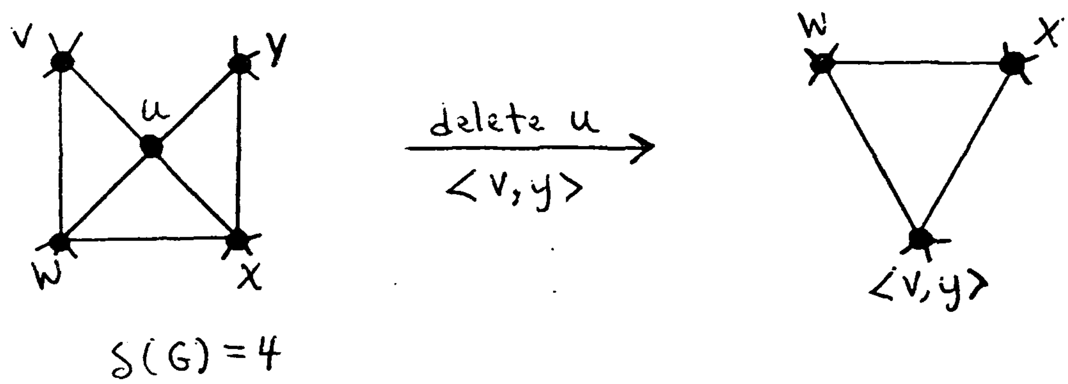
Recursive Reduction Coloring:

Key steps:

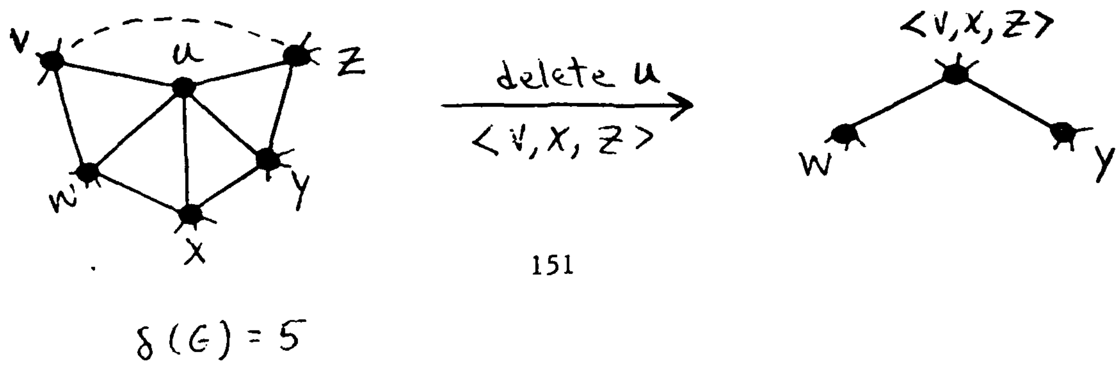
1. delete minimum degree vertex.
2. identify pairs of nonadjacent vertices.

identification of vertices u and v: $<u,v>$
--merge $A[u]$ into $A[v]$ and delete u from G.

(1)



(2)



Algorithm RNG_4color.

1. Delete minimum degree vertex u in G.
2. Stack u and identify 2 or 3 nonadjacent vertices.
3. Repeat steps 1-2 until < 5 vertices remain.
4. Assign colors 1-4 to remaining four vertices.
5. Unstack and assign colors to remaining $n-4$ vertices of G.

Run-time Analysis:

1. all deletions require $O(n)$ time.
2. all identifications require $O(n^2)$ time.
3. stacking/unstacking vertices requires $O(n)$ time.

=> $T(n) = O(n^2)$.

Performance: Successful on random RNGs and maximal RNGs with $n \leq 200$ vertices.

RNG Conjecture:

(1) The minimum degree of any RNG ≤ 4.

(2) Every vertex with degree ≤ 4 has a pair of nonadjacent neighbors with degree $\leq k$, for some constant $k < n$.

Open Problem: Can we 4-color RNGs in linear time?

4-Coloring Relatively Closest Graphs (RCGs)

Forbidden subgraph for RCGs: K_3.

Theorem 5 [Cimikowski]. $e(RCG) \leq 2n-5$.

Corollary 5.1. The minimum degree of an RCG is ≤ 3.

Reduction 4-coloring algorithm (exact):

```
while n > 4 do
      remove a vertex u with minimum degree (≤3)
          and stack u;
assign colors 1-4 to remaining 4 vertices;

while stack not empty do
      remove vertex u from stack and assign a color
         from 1-4 to u;
```

$T(n) = O(n)$.

Graphs	Chromatic Number	4-Coloring (exact)	3-Coloring (exact)
Planar graphs	NP-complete	$O(n^2)$	?
DGs	$O(n)$	$O(n^2)$	$O(n)$
RNGs	?	$O(n^2)$	?
RCGs	?	$O(n)$	?
GGs	?	$O(n^2)$	?

Table 1. Complexities of Coloring.

Improved heuristics have been found for:

(1) RNG 4-coloring.

(2) GG 4-coloring.

(3) RCG 3-coloring.

Conclusions:

1. Forbidden subgraphs and relative sparsity make proximity graphs easier to color than arbitrary planar graphs.

2. RCGs are easiest because of sparsity and minimum degree.

3. DGs are easier because of inner triangularity.

4. RNGs are somewhat easier because of sparsity and forbidden subgraphs.

4. GGs are only slightly easier (not quite as dense).

5. Good average-case algorithms may be obtainable for RNGs and GGs.

   $e_{ave} = 1.27n$ and $degree_{ave} = 2.5$ for RNGs.
   $e_{ave} = 2n$ and $degree_{ave} = 4$ for GGs.

Future Research:

1. Find Kuratowski-like characterizations for proximity graphs.

2. Investigate other hard graph problems for proximity graphs:
   - dominating sets, independent sets, edge coloring

3. Study other kinds of proximity graphs.

4. Investigate further relationships between *PFnets* and other proximity graphs.

Proximity Graphs in Computer Vision

Godfried Toussaint
McGill University
Montreal, Canada

The abstract and transparencies for this talk were not available.

Dynamic Shape Graphs of Molecules

Paul Mezey
University of Saskatchewan
Saskatoon, Canada

The abstract and transparencies for this talk were not available.

On k Relative Neighborhood Graphs

M. S. Chang*, C. Y. Tang** and R. C. T. Lee***

* M. S. Chang is with the Institue of Computer Science and Information Engineering, National Chung Cheng University, Chiayi, Taiwan, Republic of Chian.

**C. Y. Tang is with the Institute of Computer Science, National Tsing Hua University, Hsinchu, Repulic of China.

*** R. C. T. Lee is with the National Tsing Hua University, Hsinchu, Taiwan and Academia Sinica, Taipei, Taiwan, Repulic of China.

Abstract

A bottleneck optimization problem on general graphs with edge costs is the problem of finding a subgraph of a certain kind that minimizes the maximum edge cost in the subgraph. A Euclidean bottleneck optimization problem is a bottleneck optimization problem on complete graphs which are constructed from a set of points in the plane and whose edge costs are Euclidean distances between points connected by edges. In this dissertation, we define a special graph called k–Relative Neighborhood Graph , denoted as kRNG, where k is a positive number , and use it to solve the following three Euclidean bottleneck optimization problems:

(A) The Euclidean bottleneck matching problem.

(B) The Euclidean bottleneck biconnected edge subgraph problem.

(C) The Euclidean bottleneck traveling salesperson problem.

We prove the following three theorems:

(1) For any instance of Problem A , there exists an optimal solution which is a subgraph of a 17RNG.

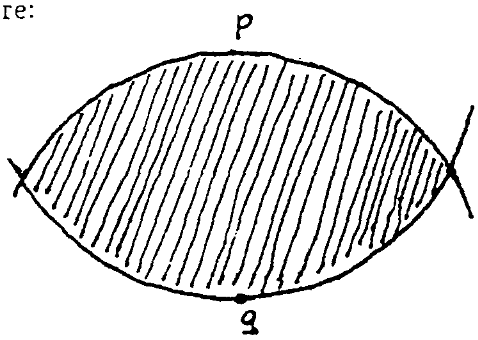
(2) For any instance of Problem B, there exists an optimal solution which is a subgraph of a 2RNG.

(3) For any instance of Problem C , there exists an optimal solution which is a subgraph of a 20RNG.

All numbers of edges of these three special graphs are $O(n)$. Therefore we can find optimal solutions for the above three problems from these three k–relative neighborhood graphs. In this way, we can solve Problem A and Problem B in $O(n^2)$ time, and also an efficient approximation algorithm for Problem C is developed. The third theorem above gives us an interesting graph theoretic result: 20RNGs are Hamiltonian.

1 The lune of two points on the plane

Let p and q be two points on the plane. Draw two circles with radius d_{pq} (Euclidean distance between p and q) centered at p and q respectively as shown in the following figure:



The shaded area (not including its boundary) is called the lune of p and q. Formally, $LUN_{pq} = \{ x \mid x \in R^2, d_{px} < d_{pq} \text{ and } d_{qx} < d_{pq} \}$.

2. k Relative Neighbors

Given a set V of points on the plane, p and q are called k relative neighbors if and only if

(i) $p \in V$ and $q \in V$,

and

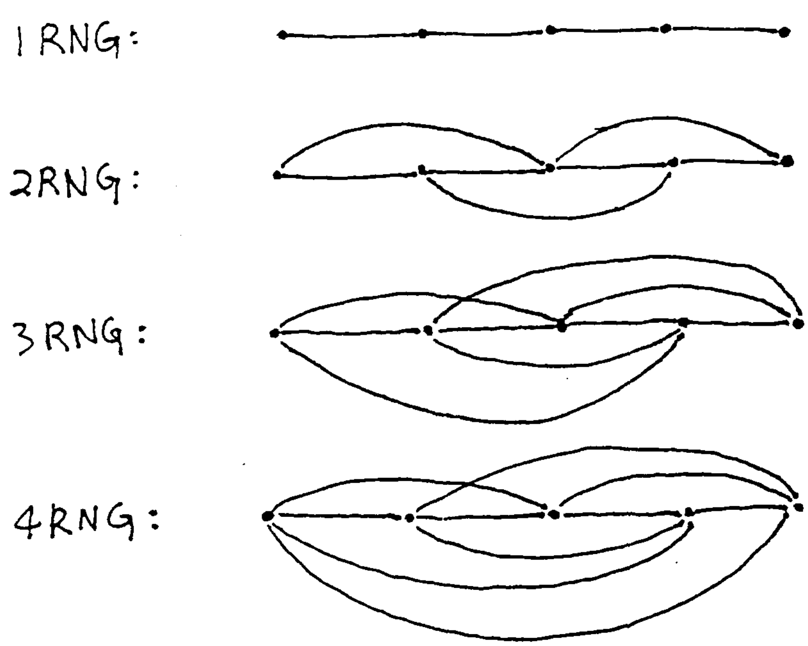
(ii) $|LUN_{pq} \cap V| < k$.

3. k Relative Neighborhood Graphs

Given a set V of points on the plane, connect two points if their lune contains less than k points of V. The graphs constructed in this way are called *k relative neighborhood graphs*. Formally, kRNG = (V,E_r) where

$$E_r = \{ (p, q) \mid p, q \in V \text{ and } |LUN_{pq} \cap V| < k \}.$$

See the following figures for examples:



4 Properties of Relative Neighborhood Graphs (RNG)

The concept of k relative neighborhood graphs is generalized from that of relative neighborhood graphs which was defined by Toussaint. In fact, 1RNG = RNG.

(1) RNG is connected.

(2) RNG is planar.

(3) There exists a minimum spanning tree which is a subgraph of RNG.

(4) There exists a bottleneck spanning tree (a spanning tree whose maximum edge cost is minimized) which is a subgraph of RNG.

(5) The number of edges of a RNG is less than $3n-6$ where $n = |V|$.

5. Properties of k Relative Neighborhood Graphs

(1) If $V > k$, then kRNG is k connected.

(2) The number of edges of a kRNG is less than $18kn$. In other words, kRNGs are sparse when k is relatively smaller than n.

(3) For $k \geq 20$, kRNGs are hamiltonian.

6. Applications of kRNG

Help to solve the following three Euclidean bottleneck optimization problems:

(1) The Euclidean Bottleneck Matching Problem.

(2) The Euclidean Bottleneck k–connected edge subgraph problem.

(3) The Euclidean Bottleneck Traveling Salesperson Problem.

7. The Euclidean Bottleneck Matching Problem

Given a set V of points on the plane, a Euclidean Bottleneck Matching is a perfect matching of V whose longest matched edge is minimized. The Euclidean Bottleneck Matching Problem is, given a set V of points, to find a Euclidean Bottleneck Matching.

Lemma: There exists a Euclidean Bottleneck Matching which is a subgraph of 17RNG.

Since 17RNG is a sparse graph, we can find a Euclidean bottleneck matching from it more quickly instead from the complete distance graph of V.

8. The Euclidean Bottleneck k–connected Edge Subgraph

Given a set V of points on the plane, we can connect it into a k–connected graph. This graph is called a Euclidean k–connected edge subgraph (a subgraph of the complete graph of V). A Euclidean k–connected edge subgraph whose longest edge is minimized is called a Euclidean bottleneck k–connected edge subgraph.

Lemma: There exists a Euclidean bottleneck k–connected edge subgraph which is a subgraph of kRNG.

Corollary: if $n > k$, then kRNG is k–connected.

9. The Euclidean Bottleneck Traveling Salesperson Problem

The Euclidean bottleneck traveling salesperson problem is to connect V into a Hamiltonian cycle such that the longest edge in the cycle is minimized. Such a cycle is called a Euclidean Bottleneck Hamiltonian Cycle.

Lemma: There exists a Euclidean bottleneck Hamiltonian cycle which is a subgraph a 20RNG.

Corollary: if $k \geq 20$, than 20RNG is Hamiltonian.

ABSTRACT

Monotonic Search Networks (MSNETs)

Govinda Kurup (Presenter)
Xerox Corporation
Webster, NY 14580

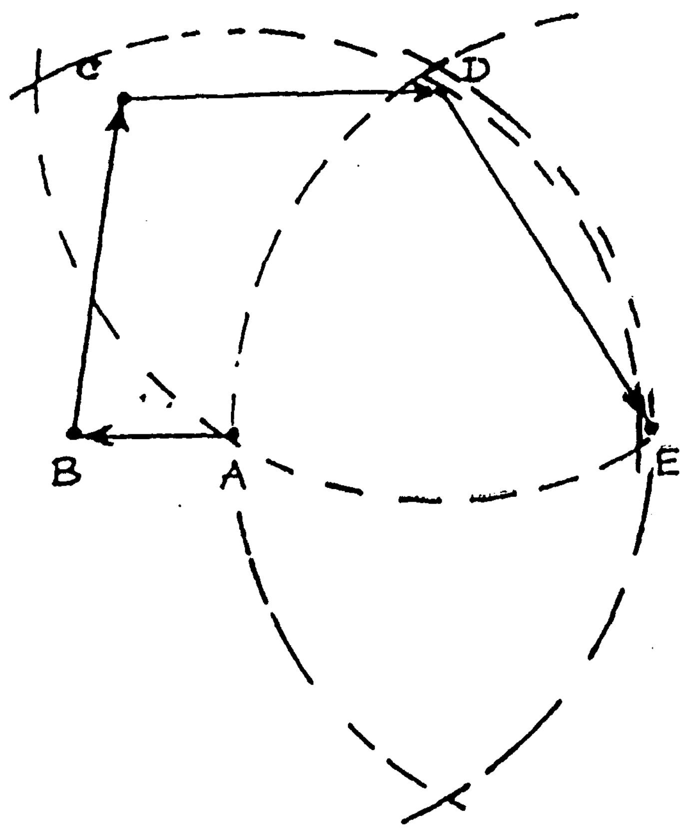
Don Dearholt
Department of Computer Science
Mississippi State University
MS 39762

A new network called a Monotonic Search Network (MSNET) is presented. In these networks, there is a monotonically decreasing distance function, and therefore a monotonic path, between every pair of nodes in the network. After discussing the foundations, an algorithm for generating the MSNET from the Relative Neighborhood graph of a set of nodes (by adding some edges) is given. An application of the MSNET as the underlying structure for an associative database for computer vision is also discussed.

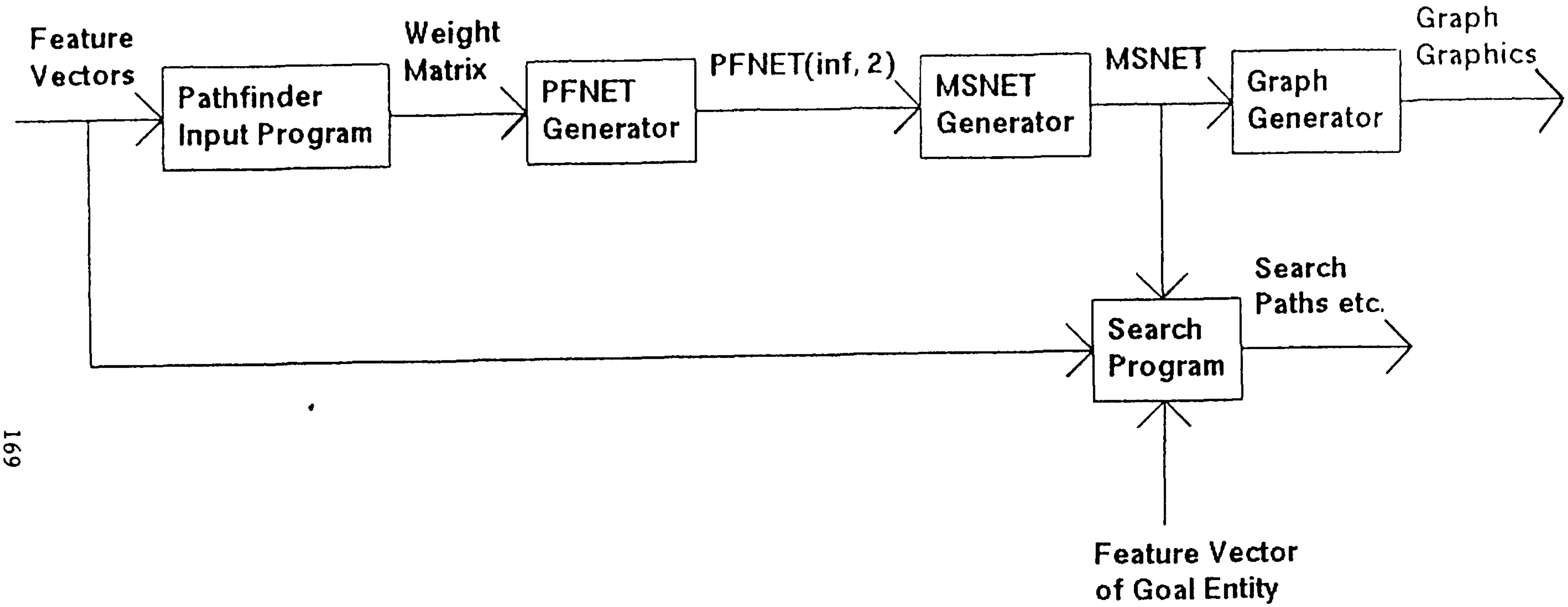
COMPUTER VISION

GOAL: SCAN THE ENVIRONMENT AND MAKE DECISIONS WITHOUT HUMAN INTERACTION

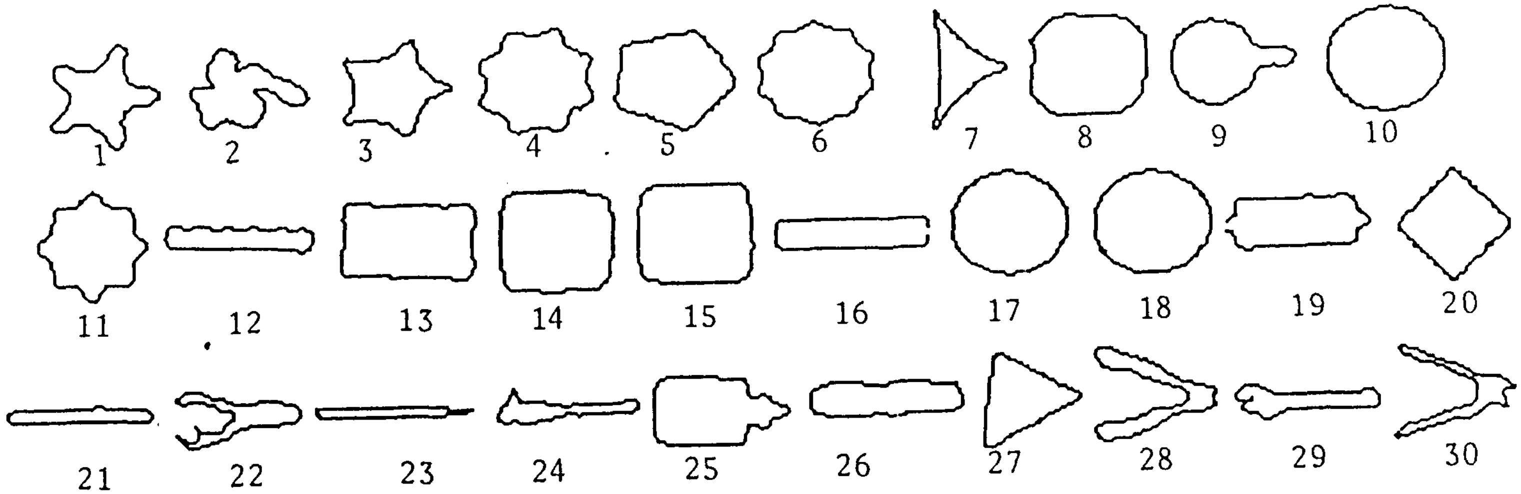
REQUIRES: KNOWLEDGE REPRESENTATION
CLASSIFICATION
ABILITY TO *DESCRIBE* SCENE
RECONSTRUCT SCENE
ENHANCE SCENE
MODIFY SCENE



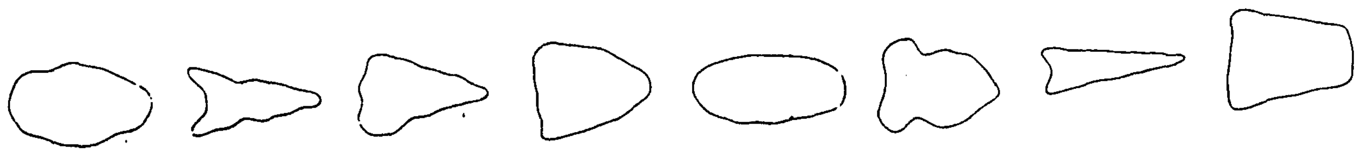
NONMONOTONIC SEARCH PATH



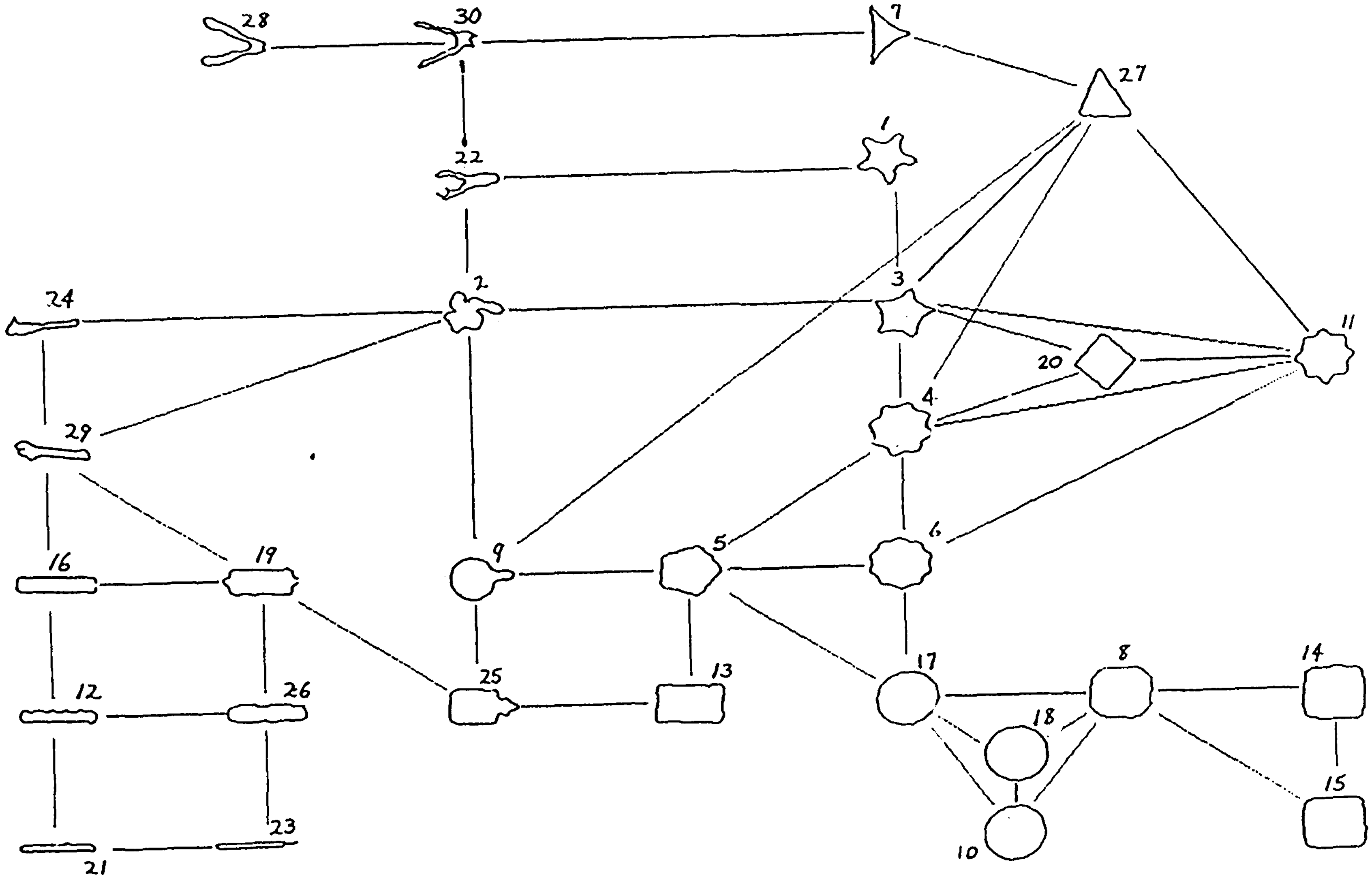
SYSTEM ORGANIZATION FOR THE DATABASE



DATABASE OBJECTS



EIGHT ARCHEOLOGICAL TEST OBJECTS



MSNET ORGANIZATION OF THE DATA

ABSTRACT

GENIE: Rapid Prototyping for Network Models

Chris Esposito
Boeing Advanced Technology Center
Seattle, Washington 98124

Networks and graphs are very often used to model a wide variety of systems and phenomena, from telecommunications networks to the organization of human semantic memory. This presentation describes a system called GENIE (General Extensible Network Interface Editor) that supports the rapid construction of many different sorts of domain-specific network models. GENIE provides basic graph-theoretic objects (nodes, edges, subgraphs, etc.) as primitive constructs and supports common attributes such as object position, color, shape, and layout/manipulation functions. An object-oriented single-inheritance extension language allows the user to attach arbitrary application-specific data structures to objects, specify graphical constraints, and provide special layout or manipulation functions. A tailorable user interface supports different interaction styles. An application interface allows GENIE to be used as a "graph server" for other applications that need a flexible graph display facility but do not want to invest the effort to develop a custom system of their own. GENIE is being implemented in C++ and Xscheme on top of X11/NeWS.

GENIE
Rapid prototyping for Network Models

Dr. Chris Esposito
Boeing Advanced Technology Center
Seattle, WA
December 2, 1989

GENIE-

General
Extensible
Network
Interface
Editor

GENIE
Introduction

Network and graph models are useful in a wide variety of areas:

- Semantic Networks
- Parse Trees
- Process Management
- Information Retrieval
- Distributed File System Management
- etc.

GENIE
Introduction

A closer look at network-model applications

1. Common functionality/data-

    position
    color
    type
    shape
    layout/manipulation
    etc.

2. Application-specific functionality/data-

    arbitrary data structures attached to objects
    special structural requirements
    special layout or manipulation functions
    differing interaction styles
    etc.

GENIE
Introduction

A closer look at network-model applications

2 different approaches to providing application-specific functionality/data:

A. Build entire application in GENIE
- self-contained system
- e.g., ANETS

B. Front-end / graph-server for other applications

- need interapplication communication
- parse trees for an NLP system

GENIE
Requirements

1. Provide a parts kit for common attributes & functions

2. Provide a means for customizing & extending GENIE

3. Provide a means for working with other applications

GENIE

GENIE Architecture

1. Display engine
2. Tailorable user interface
3. Application interface
4. Extension language
5. Construction & layout requirements

GENIE

GENIE Architecture

Display Engine

- Core system written in C++ 2.0
- X11/NeWS
- Multiple windows / "graph buffers"
- 3-D extension will probably use PEX (PHIGS Ext. to X)

GENIE
GENIE Architecture

Tailorable user interface

several ideas borrowed from GNU Emacs

- mousemaps
- keymaps
  - e.g., selection by
    - point-click
    - circling
    - selection-box

GENIE Architecture

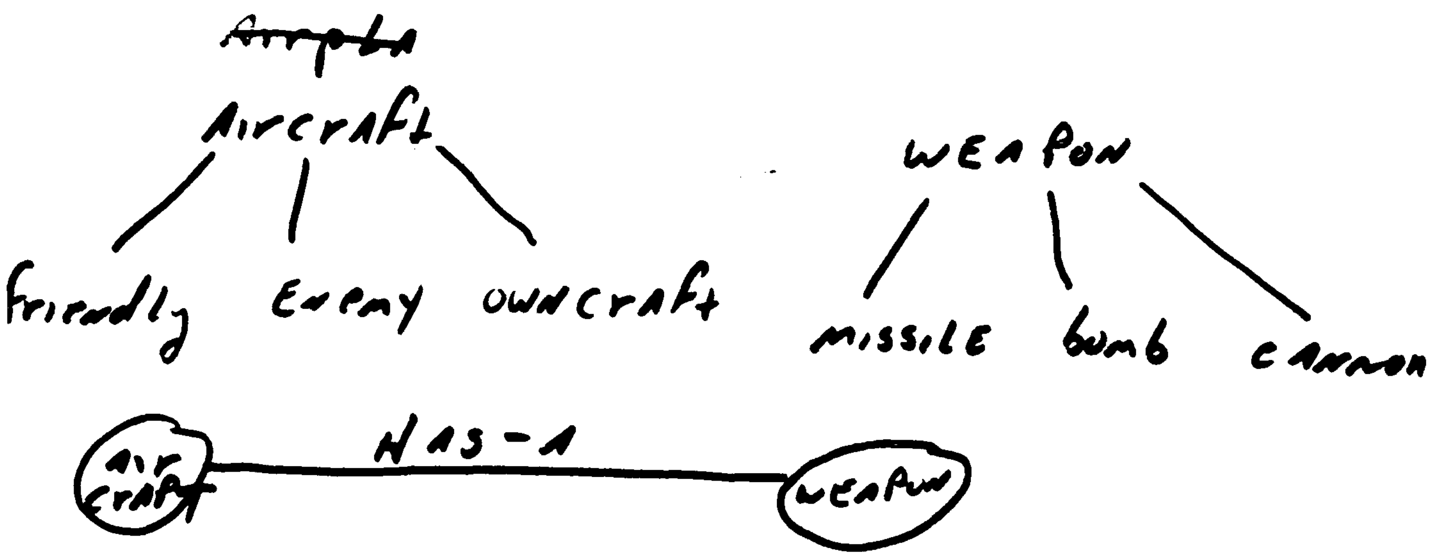
EXTENSION LANGUAGE

- LISP interpreter
- object-oriented
- SINGLE INHERITANCE
- ACCESS to C++ data structures
  functions
- Define new node/edge types
- define new manipulation functions
  layout functions
- express structural constraints

GENIE ARCHITECTURE

construction constraints –

EXAMPLE 1



EXAMPLE 2

'GATE' - type nodes -

in degree 2

out Degree 1

GENIE

GENIE Architecture

Algorithmic layout taxonomy

1. Graph class
   trees, planar, directed, etc.

2. Graphical standard
   straight lines
   grid embedding

3. Aesthetics
   minimize area
     edge crossings
     # of edge bends

4. Other constraints
   e.g., critical path nodes on straight line

5. Computational complexity
   polynomial time
   NP-hard problems
   heuristics

UNSOLVED PROBLEMS AND APPLICATIONS IN PROXIMITY GRAPHS

Donald W. Dearholt
Department of Computer Science
Mississippi State University
MS 39762

Proximity graphs offer a delightful blend of theory and applications. I shall provide a high-level description of some of the problems in which I have been engrossed in or tantalized by, partly in terms of the applications which would benefit by their solutions.

The first problem concerns the most efficient utilization of proximity graphs in the organization of data in a database intended for robotics applications. The database described by Kurup during this workshop is an example of the application of proximity graphs to this specialized database, but some issues remain unresolved. For example, the conclusion of the search process may result in any degree of match, from essentially exact to a clear mismatch; if the match is not exact, then it would be an improvement over our present capabilities (using the monotonic search network) if we knew that the search process resulted in the best match available in the database. While traversing the search path, it would also be expeditious to collect the data needed to add the new exemplar into the database efficiently, if that is desired. It is likely that a better understanding of proximity graphs will help in the solution of both of these problems.

Information retrieval, particularly in the context of a hypertext system with a graphical interface, is likely to benefit from the organization of data according to the edges in some proximity graph. The work on Pathfinder in this area shows some promise, and further refinement may be possible using a more appropriate proximity graph. Important features of this application include (1) the clustering and support of higher levels of abstraction provided by the Pathfinder networks, (2) multiple associative paths between highly related concepts, (3) effective search and browsing procedures, and (4) efficient ways of adding new information. The proximity graphs which include any Pathfinder network provide support for the first two items above; the third and fourth items on the list, however, are more difficult, particularly in the area of information retrieval in which assumptions regarding keys or semantics are involved. The use of proximity graphs may allow an approach which lies somewhere between the (relatively simple) purely syntactic and the (relatively expensive) semantic modeling approaches in both cost and performance.

The most fascinating problem, from my perspective, is the possibility of developing a unified model of some important aspects of perception and cognition. While this may sound grandiose, the Relative Neighborhood Graph provides meaningful perceptual representations of objects; the Delaunay triangulation graph is, among other things, the dual of a representation of the decision space (the Voronoi diagram) for a minimum-distance pattern classifier (which could be based on the Selfridge model called Pandemonium); and the Pathfinder networks are intended to model human associative memory. Thus it now appears conceivable to consider the possibility of a unified model for some important aspects of both cognition and perception. In this model, percepts would be represented by one type of proximity graph, say "P"; then some transformations upon "P" would generate, augment, or modify a representation for a corresponding set of concepts, represented by another related proximity graph, say "C". The system of proximity graphs used in this unified model and the transformations between graphs could provide a new perspective on the transformations of information from episodic memory to semantic memory.

SOME UNSOLVED PROBLEMS ON PROXIMITY GRAPHS*

Godfried T. Toussaint
School of Computer Science
McGill University
3480 University Street
Montreal, Quebec
CANADA H3A 2A7

ABSTRACT

Recent developments in the field of *computational morphology* (spatial and cluster analysis, computer vision, pattern recognition, computational perception, etc.) are making ever increasing use of proximity graphs. Thus it becomes increasingly relevant to understand the properties of such graphs as well to design efficient algorithms for their computation. In this note we mention some open problems in this area.

1. Computational Morphology

1.1 The Shape of a Set of Points

1.1.1 Introduction

One of the central problems in shape analysis is extracting the shape of a set of points. Let $S=\{x_1, x_2, \ldots, x_n\}$ be a finite set of points in the plane. The relative neighborhood graph (RNG) [To80a] and the β-skeletons [KR85] are two structures that have been well investigated in this context. The RNG is obtained by joining two points x_i and x_j of S with an edge if Lune(x_i, x_j) does not contain any other points of S in its interior. Lune(x_i, x_j) is defined as the intersection of the two discs centered at x_i and x_j with radius equal to the distance between x_i and x_j. One of the best known proximity graphs on a set of points is the Delaunay triangulation (DT) and it is well known that the DT is a supergraph of the RNG [To80a]. The β-skeletons are a generalization of RNG's and Gabriel graphs and the lune-based neighborhoods in question are a function of a parameter β. In [To88b] a new graph termed the *sphere-of-influence* graph is proposed as a primal sketch intended to capture the low-level perceptual structure of visual scenes consisting of dot-patterns (point-sets). The graph suffers from none of the drawbacks of previous methods and for a dot pattern consisting of n dots can be computed efficiently in O(n log n) time. For a survey of the most recent results in this area see the paper by Radke [Ra88].

---

* Presented at the *First Workshop on Proximity Graphs*, Las Cruces, New Mexico, December 1989.

1.1.2 The Relative Neighborhood Graph

In [JK89] it is shown that the RNG in 3-space can be computed in $O(n^2 \log n)$ time and $O(\mu_3(S))$ space where $\mu_3(S)$ denotes the size of RNG(S). It is an open question whether this upper bound can be improved. It is also not known how large $\mu_3(S)$ can be over all instances of S. Denote this value by $\mu_3(n)$. It is shown in [JK89] that $\mu_3(n) = O(n^{(3/2)+c})$ where c is a positive constant and they conjecture that $\mu_3(n) = O(n)$.

1.1.3 β-Skeletons

In [KR85] it was shown that lune-based β-skeletons with $\beta > 1$ could be computed in $O(n^2)$ time. In [JKY89] it is shown that lune-based β-skeletons with $1 \le \beta \le 2$ can be constructed in linear time from the Delaunay triangulation in any L_p metric. The Delaunay triangulation in any L_p metric can be computed in O(n log n) time [Le80]. It is an open question whether for $\beta > 2$ these skeletons can be computed in $o(n^2)$ time.

1.1.4 The Sphere of Influence Graph

Avis and Horton [AH85] showed that the number of edges in the sphere-of-influence graph is bounded above by 29n. The best upper bound to date is 17.5. This follows from a lemma of Bateman in geometrical extrema suggested by a lemma of Besicovitch (*Geometry*, May 1951, pp. 667-675) and an observation of Kachalski. Bateman's lemma gives 18n and Kachalski's trick reduces it by .5. The same trick reduces Avis & Horton's bound by .5. David Avis conjectures that the best upper bound is 9n.

1.2 Polygon decomposition

1.2.1 Simple polygons

The problems of decomposing simple polygons into various types of more structured polygons have a number of practical applications and have received considerable attention recently from the theoretical perspective. See [To88a] for several papers discussing recent issues. In pattern recognition it is desired to obtain decompositions into meaningful parts. The so-called *component-directed* methods decompose the polygon into well established classes of simpler polygons such as convex or star-shaped polygons. These decompositions are satisfactory from the morphological point of view only rarely. Another approach which may be superior is to use *procedure-directed* methods based on proximity graphs. In [To80b] it was proposed to use the *relative-neighbour decomposition* (RND) of a simple polygon P of n vertices and an $O(n^3)$ time algorithm for its computation was given. ElGindy and Toussaint [ET88] reduced this complexity to $O(n^2)$. Two vertices p_i and p_j of a simple polygon are relative neighbours if their lune contains no other vertices of P that are visible from either p_i or p_j. Two vertices p_i and p_j are said to be visible if the line segment $[p_i, p_j]$ lies in P. It is an open question whether this decomposition can be computed in $o(n^2)$ time and neither is a super-linear lower bound known for this problem.

1.2.2 Special classes of polygons

The fastest known algorithm [ET88] for computing the RND of a simple polygon is $O(n^2)$. On the other hand, for *convex* polygons the RND can be computed in O(n) time [Su83], and so can the Delaunay triangulation [AGSS]. However, it is shown in [ART87] that O(n log n) is a lower bound for computing the Delaunay triangulation on the vertices of a *star-shaped* or *monotone* polygon. It is unknown whether any other proximity graphs can be computed in linear time for the case of convex polygons. Furthermore, for most proximity graphs it is unknown whether they can be computed in $o(n^2)$ time for special classes of simple polygons such as *star-shaped, monotone* or *unimodal* polygons. For *unimodal* polygons the RNG and MST can be computed in O(n) time [O189]. It is unknown whether the Delaunay triangulation on the vertices of a *unimodal* polygon can be computed in linear time.

2. Recognizing Proximity Graphs

One area as yet almost totally unexplored concerns the question of the recognition of proximity graphs. The only known result concerns Delaunay triangulations. Given a triangulation T of a set of n points, Ash & Bolker [AB85] have shown that whether T is a Delaunay triangulation can be determined in O(n) time.

3. Graph Theoretic Properties of Proximity Graphs

Another area which has received little attention concerns the determination of graph theoretical properties of proximity graphs. The only proximity graphs which have been carefully examined are the Gabriel graph [MS80] and the RNG [Ur83].

4. Probabilistic Properties of Proximity Graphs

Yet another area which has received little attention concerns the determination of probabilistic and statistical properties of proximity graphs. The only proximity graphs which have been carefully examined are the Delaunay triangulation, the Gabriel graph, and the RNG. Miles [Mi70] has done considerable work on the probability distribution of random variables describing characteristics of the Delaunay triangulation. See also Getis & Boots [GB78]. Devroye [De88] obtains a variety of results concerning the expected number of edges in proximity graphs such as the Gabriel graph, the RNG and several types of nearest neighbour graphs. No results of this type are known for all the other proximity graphs discussed in this note.

5. References

[AGSS] A. Aggarwal, L. Guibas, J. Saxe, and P. W. Shor, "A linear time algorithm for computing the Voronoi diagram of a convex polygon," *Proc. 19th ACM Symposium on the Theory of Computing*, 1987, pp. 39-45.

[AH85] D. Avis and J. Horton, "Remarks on the sphere of influence graph," in *Discrete Geometry and Convexity*, Eds., J. E. Goodman et al., New York Academy of Sciences, 1985, pp. 323-327.

[ART87] A. Aggarwal, P. Raghavan, and P. Tiwari, "Lower bounds for closest pair and related problems in simple polygons," *IBM T. J. Watson Tech. Rept.*, in press.

[AB85] P. F. Ash and E. D. Bolker, "Recognizing Dirichlet tessellations," *Geometria Dedicata*, vol. 19, 1985, pp. 175-206.

[De88] L. Devroye, "The expected size of some graphs in computational geometry," *Computers & Mathematics with Applications*, vol. 15, No. 1, 1988, pp. 53-64.

[ET88] H. A. ElGindy and G. T. Toussaint, "Computing the relative neighbor decomposition of a simple polygon," in *Computational Morphology*, G. T. Toussaint, Editor, North-Holland, pp. 53-70.

[GB78] A. Getis and B. Boots, *Models of Spatial Processes: An Approach to the Study of Point, Line, and Area Patterns*, Cambridge University Press, 1978.

[JK89] J. W. Jaromczyk and M. Kowaluk, "Constructing the relative neighborhood graphs in 3-dimensional Euclidean space," manuscript 1989.

[JKY89] J. W. Jaromczyk, M. Kowaluk, and F. Frances Yao, "An optimal algorithm for constructing β-skeletons in L_p metric," manuscript 1989.

[KR85] D. G. Kirkpatrick and J. D. Radke, "A framework for computational morphology," in *Computational Geometry*, G. T. Toussaint, Ed., North-Holland, 1985, pp. 217-248.

[Le80] D. T. Lee, "Two dimensional Voronoi diagram in the L_p metric," *Journal of the ACM*, vol. 27, 1980, pp. 604-618.

[Mi70] R. E. Miles, "On the homogeneous planar Poisson point process," *Mathematical Biosciences*, vol. 6, 1970, pp. 85-127.

[MS80] D. W. Matula and R. R. Sokal, "Properties of Gabriel graphs relevant to geographic variation research and the clustering of points in the plane," *Geographical Analysis*, vol. 12, 1980, pp. 205-222.

[Ol89] S. Olariu, "A simple linear-time algorithm for computing the RNG and MST of unimodal polygons," *Information Processing Letters*, vol. 31, June 1989, pp. 243-247.

[Ra88] J. D. Radke, "On the shape of a set of points," in *Computational Morphology*, Toussaint, G. T., ed., North-Holland, 1988, pp. 105-136.

[Su83] K. J. Supowit, "The relative neighborhood graph, with an application to minimum spanning trees," *Journal of the ACM*, vol. 30, No. 3, July 1983, pp. 428-448.

[To80a] G. T. Toussaint, "The relative neighbourhood graph of a finite planar set," *Pattern Recognition*, vol. 12, 1980, pp. 261-268.

[To80b] G. T. Toussaint, "Decomposing a simple polygon with the relative neighbourhood graph," *Proceedings of the Allerton Conference*, October 1980, pp. 20-28.

[To86] G. T. Toussaint, "Computational geometry and morphology," *Science on Form: Proc. First International Symposium for Science on Form*, S. Ishizaka, Ed., KTB Scientific Publishers, Tokyo, 1986, pp. 395-403.

[To88a] Toussaint, G. T., ed., *Computational Morphology*, North-Holland, 1988.

[To88b] Toussaint, G. T., "A graph-theoretical primal sketch," in *Computational Morphology*, Toussaint, G. T., ed., North-Holland, 1988, pp. 229-260.

[Ur83] R. B. Urquhart, "Some properties of the planar Euclidean relative neighborhood graph," *Pattern Recognition Letters*, vol. 1, July 1983, pp. 317-322.

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Computing Research Laboratory New Mexico State University 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Department of the Navy Office of Naval Research
6c. ADDRESS (City, State and ZIP Code) Box 30001, Dept. 3CRL Las Cruces, NM 88003	7b. ADDRESS (City, State and ZIP Code) Code 1511: TEL 800 North Quincy Street Arlington, Virginia 22217-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Dept of the Navy, Office of Naval ~~Research~~ 8b. OFFICE SYMBOL (If applicable) 1511	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00014-90-J-1323
8c. ADDRESS (City, State and ZIP Code) 800 North Quincy Street Arlington, Virginia 22217-5000	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT NO.

11. TITLE (Include Security Classification)
Proceedings Of The First Workshop On Proximity Graphs

12. PERSONAL AUTHOR(S)
Dearholt, Donald W. and Harary, Frank N.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
Proceedings of Workshop	FROM 89-Dec-1 TO 91-May-31	91-May-31	198

16. SUPPLEMENTARY NOTATION
Proceedings Of The First Workshop On Proximity Graphs

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE NUMBER (Include Area Code) / 22c. OFFICE SYMBOL